Nr 275
1984

Prace naukowe AE

Akademii Ekonomicznej imienia Oskara Langego we Wrocławiu

ISSN 0324-8445

INFORMATYKA

WROCŁAW 1984

Nr 275 PRACE NAUKOWE

Akademii Ekonomicznej imienia Oskara Langego
1984 we Wrocławiu

INFORMATYKA

WROCŁAW 1984

ZESPÓŁ REDAKCYJNY
Janusz WOJDYŁA, Jadwiga SOBIESKA-KARPIŃSKA

RECENZENCI
Mieczysław BAZEWICZ, Ignacy DZIEDZICZAK, Zbigniew KIERZKOWSKI
Bernard KUBIAK, Elżbieta NIEDZIELSKA, Magdalena SZANIAWSKA,
Tadeusz WIERZBICKI

TŁUMACZE STRESZCZEŃ
Krzysztof JAJUGA, Jacek LEWIŃSKI

Maszynopisy pracy złożono w Wydawnictwie 23 VI 1982 r.

ISSN 0324 – 8445

WYDAWNICTWO UCZELNIANE AKADEMII EKONOMICZNEJ WE WROCŁAWIU

Nakład 150 + 22 egz. Format B5. Ark. wyd. 11,5. Ark. druk. 12. Papier offset. kl. III, 70 g, B1. Oddano do druku w listopadzie 1984 r. Druk ukończono w grudniu 1984 r. Zakład Graficzny Politechniki Wrocławskiej. Zam. nr 3651/84 – Z-6 – Cena zł 110,–

SPIS TRESCI

Elżbieta N i e d z i e l s k a, Próba klasyfikacji czynników rozwoju informatycznych systemów zarządzania 5
Andrzej M a ł a c h o w s k i, Adam N o w i c k i, Próba klasyfikacji metod i technik analizy systemów informacyjnych 19
Wojciech D o m i ń s k i, Próba kompleksowego ujęcia struktury eksploatacji informatycznego systemu zarządzania .. 33
Jerzy K o r c z a k, Projektowanie dialogu w komputerowych systemach uczących /na przykładzie systemu OSKAR/..... 43
Jerzy K o r c z a k, Barbara Ł u k a s i k - M a k o w s k a, Weryfikacja materiału dydaktycznego w komputerowych systemach uczących 57
Janusz W o j d y ł a, Interaktywne laboratorium wspomaganego komputerem projektowania baz danych 69
Mieczysław O w o c, Jolanta S o w i ń s k a, Programowanie procesu ładowania w sieciowych bazach danych 87
Marek S k w a r n i k, Analiza możliwości wykorzystania zasobów systemu zarządzania bazą danych w procesie ochrony jakości bazy 105
Mirosław D y c z k o w s k i, Analiza przydatności środków ochrony dostępu do danych USZBD RODAN dla systemu zarządzania wspomaganego komputerem 121
Colette R o l l a n d, Jan S t ę p n i e w s k i, Metoda konstrukcji relacyjnego modelu danych na przykładzie rachunkowości finansowej 137
Stanisław Z i m n o c h o, Zastosowanie interpolacji do przeszukiwania zbiorów danych 145
Adam P a p s t, Zasady doboru techniki zbierania danych dla wsadowych systemów informatycznych 159
Barbara Ł u k a s i k - M a k o w s k a, Możliwości zastosowania komputera do prac związanych z projektowaniem odzieży w przemyśle 169
Bernard K u b i a k, O pewnych metodach analizy systemów informatycznych 179

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Elżbieta N i e d z i e l s k a
Akademia Ekonomiczna we Wrocławiu

PRÓBA KLASYFIKACJI CZYNNIKÓW ROZWOJU INFORMATYCZNYCH SYSTEMÓW ZARZĄDZANIA*

Uwagi wstępne

Jednym z ważnych elementów rozwoju informatycznych systemów zarządzania gospodarczego jest zagadnienie prawidłowej klasyfikacji ogółu czynników wzrostu. Proponujemy zatem pewien schemat podziału, opatrzony dwuczłonowym komentarzem słownym zawierającym każdorazowo definicję rodzajową oraz przykład ilustracyjny.

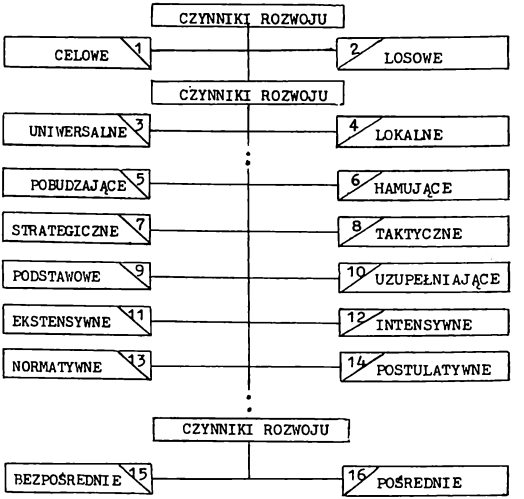
Rysunek 1 zbudowano zgodnie z zasadami tzw. klasyfikacji komasacyjnej, według której każdy z wyróżnionych poziomów jest wynikiem odrębnej czynności podziału, dokonywanej na podstawie innego kryterium, ale zawsze w stosunku do tego samego zbioru desygnatów, tj. czynników rozwoju. W związku z tym samo następstwo poziomów, a więc i kolejność komentowania poszczególnych elementów rysunku nie wpływają na istotę zagadnienia.

2. Definicje czynników rozwoju i ich przykładowe ilustracje

Określenie 1

Za czynniki c e l o w e uważa się te, które w sposób świadomy i z góry zamierzony, tj. w pełni zgodny z intencjami sprawcy, kształtują tendencje rozwoju informatycznych systemów zarządzania gospodarczego.

*Artykuł stanowi syntezę kilku wcześniejszych opracowań autorki, poświęconych szeroko ujmowanej problematyce rozwoju systemów informatycznych.



Rys. 1. Schemat klasyfikacji czynników rozwoju informatycznych systemów zarządzania

Przykład 1

Jednym z ważnych czynników o charakterze celowym jest systemowa zgodność mechanizmów rozwojowych informatyki i zarządzania jako dwu współzależnych sfer społeczno-gospodarczego rozwoju kraju. W wykładni przedmiotowej zasada zgodności wyraża pożądaną jednoznaczność uporządkowania relacji: system informatyczny - system zarządzania. Mianowicie, model systemu informatycznego powinien być zawsze pochodną modelu zarządzania i pełnić wobec niego służebną rolę realizacyjną. Wykładnię podmiotową można wyrazić natomiast następująco: strategia funkcjonowania i rozwoju systemów informatycznych powinna być zgodna ze strategią funkcjonowania i rozwoju struktur organizacyjnych tych obiektów, dla których te systemy są przeznaczone [6].

Określenie 2

Czynniki l o s o w e to takie, które formują tendencje rozwoju systemowego w sposób przypadkowy i niezamierzony, a więc niezgodny - w całości lub w części - z intencjami sprawcy. Innymi słowy, są to czynniki deformujące /odkształcające/ w jakimś stopniu oczekiwany /prognozowany - programowany - planowany/ trend rozwojowy.

Przykład 2

Ilustracją działania czynnika losowego jest obecna k r y z y s o w a s y t u a c j a g o s p o d a r c z a naszego kraju, z której wynikają pewne bezpośrednie implikacje, niekorzystne dla tempa i zakresu bieżących zastosowań informatyki. Otóż daje się zaobserwować zjawisko polaryzacji działań komputeryzacyjnych w skali makroskopowej. Polega ono na wyraźnym spadku zainteresowania informatyką u jej faktycznych i potencjalnych użytkowników, którzy niechętnie angażują się w jakiekolwiek przedsięwzięcia wiążące się z komputerami, a w skrajnych przypadkach rezygnują z eksploatacji już stosowanych systemów. Wspomnianemu spadkowi zapotrzebowania na informatykę przeciwstawia się obserwowany wzrost gotowości profesjonalistów do świadczenia różnorodnych usług: obliczeniowych, projektowo-programowych, marketingowych, serwisowych, doradczych, szkoleniowych itp. Przedłużające się jaskrawe niezrównoważenie popytu i podaży informatycznej może wywoływać nie tylko dalsze zakłócenia w postępie systemowym, lecz także znacząco deformować długookresowy, prognozowany trend rozwoju [4].

Określenie 3

Pojęcie czynników u n i w e r s a l n y c h odnosimy do takich elementów, które wyznaczają prawidłowości rozwoju systemów informatycznych niezależnie od czasu i miejsca ich występowania.

Przykład 3

Skuteczność komputeryzacji zależy wprost od zaakceptowania jej form przez obu uczestników tego procesu, tj. zarówno profe-

sjonalistów, jak i użytkowników. Dlatego formowanie prawidłowych więzi kooperacyjnych między specjalistycznymi jednostkami, oferującymi usługi informatyczne, oraz podmiotami społeczno-gospodarczymi różnych szczebli, pożytkującymi te świadczenia, uznajemy za czynnik uniwersalny, który w każdych warunkach pozytywnie oddziałuje na rozwój systemu. Omawiana prawidłowość rozwoju informatyki, a mianowicie przesunięcie akcentów z preferowanych poprzednio uwarunkowań, dotyczących sprzętu i programów /hardware, software/, na czynnik ludzki /peopleware/, występowała w Polsce na przełomie lat siedemdziesiątych i osiemdziesiątych [4].

Określenie 4

Czynniki lokalne w odróżnieniu od omówionych poprzednio zachowują swe znaczenie i aktualność sprawczą wyłącznie w odniesieniu do określonych warunków czasowo-przestrzennych rozwoju systemów informatycznych.

Przykład 4

Na przełomie lat siedemdziesiątych i osiemdziesiątych ujawniło się jeszcze inne zjawisko, które wpłynęło na proces rozwoju zastosowań informatyki, ale tylko w pewnych specyficznych warunkach zarządzania gospodarczego. Chodzi o społeczną aprobatę wszelkich poczynań związanych z komputeryzacją, która wymaga przecież niemałych zasobów finansowych, materiałowych, kadrowych itp. Ich wykorzystanie, na przykład w okresie przejściowego zachwiania równowagi gospodarczej, nabiera natychmiast wymowy społecznej. Tym samym dalszy rozwój informatyki, a więc wybór priorytetowych kierunków badań naukowych, planowane tempo wzrostu produkcji sprzętu, zmiana kierunków zastosowań czy wreszcie lokalne przesunięcia zatrudnieniowo-kadrowe w branży komputeryzacyjnej będą w dużej mierze uzależnione od stopnia uznania wszystkich tych rodzajów działalności nie tylko za korzystne ekonomicznie, lecz także - a może nawet przede wszystkim - za użyteczne społecznie [5].

Określenie 5

Za czynniki p o b u d z a j ą c e rozwój systemów informatycznych uważamy takie, które stymulują procesy rozwoju w pożądanym kierunku. Można tu wyróżnić aktywatory, tj. czynniki wywołujące /inicjujące/ samo zjawisko rozwoju, oraz akceleratory, tj. czynniki przyspieszające ten rozwój.

Przykład 5

Dobrą ilustracją czynników pobudzających wzrost informatycznych systemów zarządzania gospodarczego jest i n n o w a c y j n o ś ć m e t o d y c z n a polegająca na wprowadzaniu nowatorskich zasad i narzędzi projektowo-programowych tylko korygujących bądź również doskonalących pierwotne rozwiązania architektury systemowej. Samo pojęcie innowacji może mieć dwojaką wykładnię. W interpretacji węższej odnosi się do zabiegów przeciwdziałających fizycznemu lub moralnemu starzeniu się systemów, powodującemu ich dysfunkcjonalność eksploatacyjną. W interpretacji szerszej natomiast innowacje metodyczne są już traktowane jako element zdolny nie tylko inicjować, lecz wręcz przyspieszać wielopłaszczyznowe procesy wzrostu systemowego. O tym, czy dana innowacja informatyczna będzie pełnić wspomnianą rolę aktywatora czy też akcelatora, zadecydują w praktyce między innymi takie cechy, jak: stopień złożoności innowacji, możliwość jej modyfikowania i tzw. status oryginalności [6].

Określenie 6

Za czynniki h a m u j ą c e będziemy uważać te, które skutecznie przeciwstawiają się wytyczonemu rozwojowi procesów i zjawisk informatycznych, zachodzących w obrębie systemów zarządzania. Niektóre z tych hamulców mają działanie tylko opóźniające, inne wstrzymujące przejściowo. Są jednak wśród nich i takie, które są w stanie całkowicie powstrzymać rozwój procesów wzrostu.

Przykład 6

Do zagadnienia ilustracji hamulców rozwoju systemowego można podejść dwojako. Mianowicie, uznać, że brak wymienionych poprzed-

nio elementów pozytywnych: zgodności mechanizmów rozwojowych czy więzi kooperacyjnej, dezaprobata społeczna, jak też zastój w metodach i technikach informatycznych musi okazać się w praktyce naturalnym hamulcem wzrostu. Nam chodzi jednak o taki czynnik, który niejako z definicji, sam w sobie jest znaczącym elementem zdolnym przeciwdziałać postępowi komputeryzacji. Mamy tu na myśli niezrównoważenie bilansu siły roboczej w skali makroskopowej. Otóż, zarówno zjawisko nadmiaru wolnych rąk do pracy w ogóle, jak też ogólne zmiany w strukturze zatrudnienia /wszelkie migracje międzysferowe na dużą skalę/ przeczą idei komputeryzacji, upowszechnianej - między innymi - z myślą o wydatnej oszczędności pracy żywej. Wspomniane procesy mają na ogół charakter okresowy, a więc co najwyżej przejściowo opóźniający rozwój zastosowań informatyki. Są jednak sytuacje, w których wspomniane zjawisko nabiera cech trwałej "nieprawidłowości" ekonomicznej. Takim przykładem jest bezrobocie w rozwijających się krajach Trzeciego Świata, które okazuje się niezwykle skutecznym, doraźnym i chyba też perspektywicznym hamulcem ekspansji informatycznej [6].

Określenie 7

Czynniki s t r a t e g i c z n e - zaliczamy do nich wzajemnie niesprzeczne /zgodne/ elementy wzrostotwórcze informatycznych systemów zarządzania, wytyczające główny tor przebiegu procesów rozwoju, łącznie z celem, zakresem, metodami i środkami ich osiągnięcia, w wydłużonym horyzoncie czasowym.

Przykład 7

Działanie omawianego czynnika wygodnie jest zilustrować jedną z tzw. makrozasad umiarkowanego rozwoju zastosowań informatyki, dotyczącą s t a b i l i z a c j i teoretyczno-badawczych oraz użytkowych o s i ą g n i ę ć komputeryzacji. Otóż, są warunki, w których z jakichś powodów niemożliwe jest, nawet w dalekiej przyszłości, tworzenie tzw. generacji systemowych w wyniku naturalnego, "pełzającego" przechodzenia każdej odmiany eksploatacyjno-użytkowej systemu w kolejną udoskonaloną wersję. Wtedy za rozwój można uznać samą kontrolę, weryfikację, korektę

i modernizację stosowanych rozwiązań systemowych, a więc ogólnie: eliminowanie negatywnych i wzmacnianie pozytywnych oraz utrwalanie dotychczasowego dorobku informatycznego w tym zakresie [4].

Określenie 8

Czynniki t a k t y c z n e rozwoju informatycznych systemów zarządzania gospodarczego to takie, które zapewniają kontynuowanie, a jednocześnie dezagregowanie kierunków rozwiązań strategicznych pod względem następstwa czasowego oraz przestrzenno-osobowej strukturalizacji podejmowanych działań.

Przykład 8

L o k a l i z a c j ę o r g a n i z a c y j n ą działalności komputeryzacyjnej w dowolnym obiekcie gospodarczym uważamy za kolejną determinantę rozwoju systemów informatycznych, jednakże o znaczeniu taktycznym. Chodzi tu o odpowiedni sposób umieszczenia w strukturze danej jednostki /podporządkowanie całkowite, częściowe, wyraźna autonomia itp./ tej komórki organizacyjnej, która prowadzi całość spraw komputerowego przetwarzania danych. Przegląd rozwiązań komputerowych i praktycznych wskazuje, że stosowany wzorzec lokalizacyjny nie jest czymś stałym, podlega on bowiem swoistej ewolucji czasowo-przestrzennej, uzależnionej od profilu działalności obiektu, potrzeb informacyjnych użytkownika, specyfiki środowiska czy wreszcie priorytetów centrum w sprawach doraźnych lub regionalnych zastosowań informatyki [6].

Określenie 9

Do p o d s t a w o w y c h czynników rozwoju systemów informatycznych należą stymulatory, które mają decydujący wpływ na przebieg /rozmiar, tempo/ procesów rozwojowych. W skrajnych przypadkach ich nieobecność może całkowicie zatrzymać wspomniane procesy.

Przykład 9

Rolę czynników podstawowych dobrze ilustruje zasada s e l e k c j i dziedzin z a s t o s o w a ń systemów informatycznych. Otóż prawidłowy sposób wyznaczania głównego trendu wzrostowego komputeryzacji, wyrażony w praktyce trafnym wyborem priorytetów realizacyjnych, w każdych warunkach rozwoju gospodarki ma dla niej znaczenie decydujące. Selekcję musi jednak poprzedzać inwentaryzacja i analiza programów zastosowań, diagnoza stanu faktycznego i weryfikacja zbioru dopuszczalnych zamierzeń rozwojowych, kończyć zaś - budowa optymalnej krótkookresowej bądź nawet długookresowej prognozy wzrostu. Rezygnacja natomiast z podanej drogi postępowania oznacza rozwój żywiołowy, niekontrolowany, który na przykład w skrajnie niekorzystnej dla informatyki sytuacji ekonomicznej może przeobrazić się okresowo w swe przeciwieństwo, a więc stagnację czy wręcz regres niektórych ważnych dziedzin zastosowań [4].

Określenie 10

Czynniki u z u p e ł n i a j ą c e są to dodatkowe stymulatory procesów wzrostu. Ich obecność nie jest konieczna, mają wpływ różnokierunkowy, często kompensujący się. Czynniki te mogą wywoływać, co najwyżej, fragmentaryczne /pozytywne lub negatywne/ odkształcenia trendu rozwojowego.

Przykład 10

Za jeden z czynników uzupełniających możemy uznać s t a n d a r y z a c j ę p r z e d s i ę w z i ę ć informatycznych w szerszej klasie zastosowań. W szczególności chodzi o pożądane ujednolicenie: technik projektowania i oprogramowania systemów, reżimów organizacji ich wdrażania, zasad ich użytkowej eksploatacji, sposobów rozliczania usług obliczeniowych, metod pomiaru wydajności i szacowania efektów oraz ujednolicenie form i zawartości dokumentacji systemowej. Różnorodne standardy informatyczne mogą okazać się czasami bardzo pomocne w prawidłowym rozwoju zastosowań komputerów. Nie ma jednak dowodów na to, że na przykład ich brak jest wyraźnym hamulcem postępu informatycznego [2, s. 79].

Określenie 11

Czynniki e k s t e n s y w n e to takie czynniki sprawcze procesów rozwoju systemów informatycznych, które poziom ilościowych wskaźników wzrostu systemowego uzależniają li tylko od stopnia zwiększenia nakładów globalnych /finansowych, rzeczowych, osobowych itp./. Czynniki, o których mowa, szczególnie mocno oddziaływały w początkowym okresie zastosowań informatyki w zarządzaniu gospodarczym.

Przykład 11

K o n c e n t r a c j a sprzętowych, projektowo-programowych oraz kadrowych z a s o b ó w informatycznych niech posłuży za ilustrację czynników ekstensywnych. Ich rolę należy widzieć dwojako. W sensie bezpośrednim, jako że każdy rozwój komputeryzacji jest determinowany wprost wielkością globalnych nakładów finansowych, rozmiarem zużytego czasu, skalą zaangażowanego potencjału specjalistów, stopniem gotowości instalacyjno-eksploatacyjnej przyszłych użytkowników itd. Wykładnia pośrednia natomiast, charakterystyczna dla bardziej rozwiniętych informatycznie podmiotów komputeryzacji, może odnosić się na przykład tylko do pewnych specyficznych poczynań, rozszerzających domniemane pole przyszłych, korzystnych jej zastosowań. Działania te mogą - dla przykładu - polegać na zmianach w terytorialnym rozmieszczeniu trójzasobów, a więc przesunięciach wyposażenia sprzętowego ośrodków, ich scalaniu, redystrybucji nadwyżek mocy obliczeniowej w regionie czy też zmianach w ich użytkowaniu, gdzie chodzi - między innymi - o ujednolicenie sposobów finansowania oraz rozliczania usług informatycznych w skali makroskopowej, a także propagowanie wydajniejszych metod opracowania systemów informatycznych [4].

Określenie 12

Czynniki i n t e n s y w n e to te determinanty rozwoju systemowego, które z góry odstępują od koncepcji angażowania dodatkowych elementów wzrostotwórczych na korzyść preferowania rozwiązań wzmacniających doraźne bądź docelowe, globalne efekty pożytkowania istniejących zasobów.

Przykład 12

Jednym z ważnych czynników intensyfikacji zastosowań komputerów w procesach zarządzania gospodarczego jest k r o t n o ś ć r e a l i z a c y j n a systemów informatycznych, oparta na idei wytwarzania i rozpowszechniania tzw. systemów powtarzalnych. Próby opracowania systemów wielokrotnego użytkowania, oparte na koncepcjach banków informacji o systemach informatycznych oraz marketingu oprogramowania dowodzą niezbicie, że może to być bardzo skuteczny sposób redukowania nakładów /finansowo-materiałowych, osobowo-czasowych/ na zaplanowany rozwój komputeryzacji gospodarki [1].

Określenie 13

Czynniki n o r m a t y w n e - miano to zarezerwujemy dla tych bodźców systemotwórczych, które wywołują zmiany jakościowe i ilościowe w strukturze bądź funkcjach informatycznych systemów zarządzania.

Przykład 13

Stała dążność do zachowania r ó w n o w a g i g e n e r a c y j n e j w rozwoju informacyjnych oraz informatycznych systemów zarządzania jest - naszym zdaniem - trafną ilustracją oddziaływania elementów normatywnych. Otóż, zachowanie pewnej odpowiedniości strukturalnej i funkcjonalnej obu systemów gwarantuje unormowany rozwój wewnętrzny i zewnętrzny kolejnych eksploatacyjno-użytkowych generacji systemowych. Umożliwia on sterowanie i kontrolowanie zgodności ilościowych i jakościowych parametrów poszczególnych kategorii systemów informacyjnych i informatycznych. Tylko częściowa /ograniczona/ odpowiedniość ich struktury i funkcji, wywołana nierównomiernym rytmem wzrostu, oznacza natomiast niekorzystne naruszenie równowagi generacyjnej, a tym samym większe lub mniejsze odstępstwo od normatywów realizacyjnych [2, s. 116].

Określenie 14

Czynniki p o s t u l a t y w n e to te, które implikują zmiany strukturalno-funkcjonalne systemów w pożądanym kierunku, tempie czy zakresie, jednakże oczekiwane rezultaty tych

zmian nie muszą być identyfikowane ze zmianami standardowymi /unormowanymi/.

Przykład 14

W odróżnieniu od zilustrowanej klasy czynników normatywnych e l a s t y c z n o ś ć t e c h n o l o g i c z n a jest przykładem czynnika postulatywnego. Omawiana cecha wyraża stopień przydatności rozwiązań projektowych, programowych, eksploatacyjnych itp. do rozmaitych modyfikacji wynikających bezpośrednio z postępu w technologii przetwarzania danych. Rezultaty zmian wywołanych ewolucyjnym przekształceniem kolejnych odmian przetwarzania /na poziomie systemu operacyjnego, systemu zarządzania zbiorami danych, bazą danych oraz teleprzetwarzaniem/ nie dają się zazwyczaj ująć ani w jednolite kanony metodyczne, ani też użytkowe. Zachowują w istocie charakter zmian postulatywnych, przybliżających rozwiązania docelowe [7].

Określenie 15

Za czynniki b e z p o ś r e d n i e będziemy uważać te elementy, które odnoszą się wprost do rozwoju systemowego, ujmowanego w różnych aspektach wewnątrzwzrostowych.

Przykład 15

Jednym z bezpośrednich czynników rozwoju zastosowań informatyki jest p o t e n c j a ł o b l i c z e n i o w y komputerów, wprzęgniętych w procesy eksploatacji systemów, pracujących na potrzeby zarządzania gospodarczego w makro- i mikroskali. Oczywiście, warunek dysponowania odpowiednim rozmiarem mocy obliczeniowej to znacznie więcej niż wymaganie posiadania dostatecznej ilości technicznych środków informatyki. Konieczna jest odpowiednia koncentracja tych środków w ramach dostosowanej doń sieci wyspecjalizowanych jednostek organizacyjnych. Innym, substytucyjnym rozwiązaniem w tym zakresie jest koncepcja zindywidualizowanych zasobów mocy obliczeniowej, pozostającej w dyspozycji tych użytkowników, którzy pod względem realizacji własnych zadań przetwarzania chcą i mogą być samowystarczalni [3].

Określenie 16

P o ś r e d n i e czynniki rozwoju to takie, które wprawdzie dotyczą elementów kwalifikujących je do miana postępu informatycznego w rozwiązaniach systemowych, jednakże czynią to przez wpływ otoczenia /środowiska/ informatycznego.

Przykład 16

Czynnikiem, który sam nie stanowi sfery przetwarzaniowej, lecz oddziałuje na jej uruchamianie, a zwłaszcza na kierunki i tempo jej rozwoju, jest m i ę d z y n a r o d o w a w s p ó ł p r a c a i n f o r m a t y c z n a. Ważne są tu powiązania z krajami rozwiniętymi, zapewniające obustronną wymianę handlową, naukowo-techniczną, kulturalną itp. Przynależność do międzynarodowej kooperacji informatycznej, na przykład w dziedzinie przekazywania bieżących serwisów informacyjnych za pomocą nośników maszynowych i środków transmisji czy też posługiwanie się pewnymi standardami obowiązującymi w zautomatyzowanych systemach obsługi światowego ruchu pasażerskiego, jest ważnym bodźcem rozszerzenia skali i wzrostu natężenia procesów komputeryzacji [3].

3. Uwagi końcowe

Dokonany przegląd czynników rozwoju systemu nie pretenduje do miana wyczerpującego, a zamieszczone przykłady jedynie ilustrują działania analizowanych elementów wzrostotwórczych. Wydaje się, że przyjęta forma katalogu określeń i przykładów jest dostatecznie syntetycznym, a zarazem komunikatywnym sposobem informowania o najważniejszych aspektach zagadnienia.

LITERATURA

[1] E. N i e d z i e l s k a, Cykl życia jako teoria budowy systemu informatycznego, w: Informatyka. Teoria. Praktyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 183, Wrocław 1981.

[2] E. N i e d z i e l s k a, Funkcjonowanie i rozwój informatycznych systemów zarządzania. Koncepcja. Teoria. Praktyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 113, Wrocław 1977.

[3] E. N i e d z i e l s k a, Informatyka w zarządzaniu gospodarką narodową,[w:] Informatyka. Wstęp i technologia przetwarzania danych, red. E. Niedzielska, Warszawa 1983, PWE.

[4] E. N i e d z i e l s k a, Makrozasady umiarkowanego rozwoju zastosowań informatyki. Materiały na kursokonferencję naukową "Informatyka w warunkach reformy gospodarczej". Poznań 1982, TNOiK.

[5] E. N i e d z i e l s k a, O naprawie rzeczypospolitej informatycznej. Materiały na konferencję naukową "Ekonomika zastosowań informatyki", Wrocław 1981, PTE.

6 E. N i e d z i e l s k a, Szkice z teorii rozwoju systemów informatycznych. Determinanty rozwojowe, w: Informatyka. Teoria. Praktyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 235, Wrocław 1983.

7 E. N i e d z i e l s k a, Wprowadzenie, w: Technologia przetwarzania danych, red. E. Niedzielska, Akademia Ekonomiczna we Wrocławiu, Wrocław 1982.

A PROPOSAL FOR CLASSIFYING MIS DEVELOPMENT FACTORS

Summary

A diagram for classifying MIS development factors is analysed. The analysis involves two parts: first, a definition and second, an example of each of the following 16 development factors:

1,2. purposive and random,
3,4. general and local,
5,6. stimulative and regressive,
7,8. strategic and tactical,
9,10. basic and complementary,
11,12. extensive and intensive,
13,14. normative and postulative, and
15,16. direct and indirect.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Andrzej Małachowski, Adam Nowicki
Akademia Ekonomiczna we Wrocławiu

PRÓBA KLASYFIKACJI METOD I TECHNIK ANALIZY SYSTEMÓW INFORMACYJNYCH

1. Wprowadzenie

W teorii i praktyce doskonalenia oraz tworzenia systemów gospodarczych powstało wiele różnorodnych metod i technik analizy systemów informacyjnych. Ich stosowanie zależy od ustalonych celów badawczych.

Różnorodne działania zmierzające do usprawniania istniejących bądź projektowania nowych systemów informacyjnych komplikuje dobór właściwych metod i technik analizy. W konsekwencji utrudnia to zrealizowanie efektywnych zamierzeń modernizacyjno-projektowych informatycznych systemów zarządzania. Ponadto obserwuje się częste mylenie i nierozróżnianie metod od technik analizy. Niekiedy mówi się o metodzie zamiast o technice i odwrotnie[1].

W naszych rozważaniach przez metodę analizy będziemy rozumieć postępowanie zmierzające do usprawnienia już funkcjonującego systemu lub powstania nowego. Przez techniki analizy będziemy natomiast rozumieć środki realizacji wykorzystywane w określonej metodzie analizy. W tym ujęciu techniki analizy służą do realizacji czynności wynikających z przyjętej metody analizy.

W literaturze przedmiotu spotykamy się z dość swobodnie traktowaną i niespójną klasyfikacją metod i technik informatycznych, w tym służących analizie systemów informacyjnych[2].

W celu usunięcia trudności, hamujących rozwój teorii i praktyki analizy, podjęto próbę uporządkowania metod i technik analizy. Dokonano zatem podziału zbioru metod i technik analizy na mniej liczne grupy typologiczne.

[1] Dokładną analizę pojęć terminów "metody" i "technika" przeprowadza J. Zieleniewski [32, s. 80]. Patrz również [29].

[2] Patrz np. [10, 25].

Przeprowadzona klasyfikacja, z natury rzeczy, nie jest zupełna, może jednak ułatwiać dalsze badania oraz umożliwiać właściwy dobór metod i technik analizy w praktyce. W przyjętej klasyfikacji wyodrębniono i pogrupowano dość powszechnie znane metody i techniki analizy. Ze względu na zwięzłość rozważań podczas omawiania i charakteryzowania poszczególnych grup metod i technik ograniczono się jedynie do podania ich podstawowych cech i właściwości. Na temat wyróżnionych grup metod i technik istnieje bogata literatura. Jej podstawowy zestaw podano w spisie bibliograficznym w końcowej części niniejszego artykułu.

W przeprowadzonej klasyfikacji metod i technik analizy wykorzystano graf o podobnej strukturze. Ze względu na typ grafu klasyfikacja ta nosi nazwę klasyfikacji komasacyjnej [19].

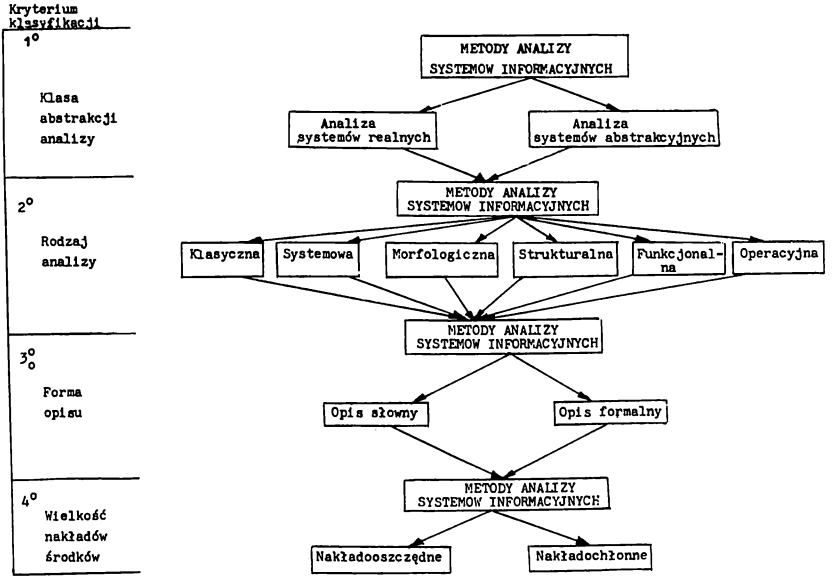
2. Klasyfikacja metod analizy

Ogólnie, metody analizy systemów informacyjnych mają dwojaki rodowód. Znaczną grupę tych metod adaptowano do potrzeb analizy systemów informacyjnych z wielu różnorodnych, niekiedy odległych od siebie dyscyplin naukowych, np. cybernetyki, statystyki, ekonometrii, teorii języków, biologii, psychologii, socjologii, nauk technicznych. Drugim źródłem był rozwój teorii analizy systemów informacyjnych i pokrewnych dyscyplin informatycznych [15], jak teoria procesów przetwarzania informacji, teoria systemów informatycznych, teoria systemów komputerowych, teoria struktur danych, teoria programowania komputerów, teoria symulacji cyfrowej, sztuczna inteligencja, organizacja zastosowań oraz ekonomika zastosowań informatyki, zagadnienia społeczno-prawne informatyki i historia informatyki.

Niezależnie od ich rodowodu w prezentowanej klasyfikacji wyodrębniono cztery podstawowe kryteria ich podziału /patrz rys. 1/.

W pierwszym członie klasyfikacji, utworzonym na podstawie kryterium klasy abstrakcji analizy wyróżniono dwie grupy metod, a mianowicie metody analizy systemów realnych i metody analizy systemów abstrakcyjnych.

Pierwsza grupa metod koncentruje się na wyszukiwaniu negatywnych właściwości badanego, rzeczywistego systemu. Badania te prowadzone są najczęściej w dwojaki sposób. Pierwszy, przez podzielenie badanego systemu na elementy składowe i rozpatrywanie każdego z nich z osobna oraz prowadzenie analizy struktury ich



Rys. 1. Strukturalny graf klasyfikacji metod analizy systemów informacyjnych

wzajemnych powiązań. Drugi, przez traktowanie systemu jako niepodzielnej całości.

Punktem wyjścia drugiej grupy metod są systemy abstrakcyjne będące wytworem twórczej wyobraźni podmiotu analizującego. Metody te, opracowane w wyniku licznych doświadczeń i poszukiwań określają racjonalny sposób podejścia do stworzenia nowych koncepcji. Wspólna właściwość tych metod polega na tym, że w pierwszej kolejności formułowana jest syntetyczna koncepcja całości, po której następuje analiza w postaci badań szczegółowych. Po nich wraca się znów do syntezy w postaci uogólnienia wniosków potwierdzających, obalających lub modyfikujących wstępną koncepcję nowego systemu [5, s. 125].

Drugim kryterium klasyfikacji jest rodzaj analizy. Wymienimy jej rodzaje: klasyczna, systemowa, morfologiczna, strukturalna, funkcjonalna oraz operacyjna.

Scharakteryzujemy te metody w kolejności zgodnej ze schematem podanym na rys. 1.

Analiza k l a s y c z n a, tzw. diagnostyczna. Charakteryzuje się tym, że jej podstawę stanowi szczegółowe badanie zmierzające do pełnej identyfikacji realnych systemów [20]. Identyfikację poprzedza charakterystyka problemu, po której podaje się cel i określenie przedmiotu badania. Materiał faktograficzny, zgromadzony za pomocą odpowiednich technik badawczych poddaje się analizie. Wnioski wyciągnięte z tej analizy podlegają ocenie, która z kolei stanowi podstawę do ustalenia zmian w systemie informacyjnym. Krótko mówiąc, w diagnostycznej metodzie analizy występują trzy etapy: analiza - synteza - ocena [22].

Analiza s y s t e m o w a. W analizie tej stosowane są dwa podejścia do badanego systemu /realnego bądź abstrakcyjnego/. System traktuje się jako niepodzielną całość lub dzieli się go na elementy składowe oraz określa się strukturę i charakter połączeń między poszczególnymi elementami. Odpowiednio bada się funkcjonowanie systemu jako całości i nie wnika w strukturę jego elementów. Określa się przy tym funkcjonowanie wszystkich elementów składowych zmierzających do ustalenia obrazu systemu [5, 30]. W odróżnieniu od metod analizy klasycznej etapy analizowania metodą systemową mają inną kolejność: synteza - analiza - ocena[3]. Ważną grupą metod tej analizy są metody prowadzące do budowy wzorca idealnego systemu [5, s. 132].

Analiza m o r f o l o g i c z n a polega na badaniu i obserwowaniu podstawowych cech i właściwości systemu, przy czym system może być traktowany w całości bądź w rozbiciu na poszczególne elementy[4]. W tym typie analizy nie prowadzi się badania struktury i własności powiązań między poszczególnymi elementami. Całą natomiast uwagę skupia się na pełnym opisie pozostałych własności i cech systemu bądź jego elementów składowych. W tej metodzie nie dąży się do ustalenia współzależności i korelacji między wartościami poszczególnych cech systemu [6, 19].

Analiza s t r u k t u r a l n a. Podstawą tego typu analizy jest w każdym przypadku podział systemu na elementy składowe i badanie struktury powiązań między tymi elementami. W badaniach tych dąży się przede wszystkim do ustalenia zgodności struktury systemu z celami systemu [6].

Analiza f u n k c j o n a l n a polega na badaniu funkcjonowania systemu zmierzającego do zrealizowania podstawowych zadań. Badania skupia się na ustaleniu wzajemnego oddziaływania elementów system - otoczenie i/lub współdziałania elementów składowych systemu w realizacji tych zadań [6;19, s. 77].

Analiza o p e r a c y j n a. Przedmiotem tej analizy jest model matematyczny lub zbiór modeli opisujących funkcjonowanie systemu realnego bądź abstrakcyjnego. Należy podkreślić, że w analizie tej niezwykle ważnymi zagadnieniami są budowa i weryfikacja modelu. O podstawowych cechach i własnościach systemu wnioskuje się na podstawie eksperymentu z modelem. W tym celu korzysta się z technik symulacyjnych. Narzędzia stosowane w tej analizie wywodzą się z badań operacyjnych. Na temat tych badań istnieje obszerna literatura przedmiotowa, także w języku polskim[5].

Kolejnym, trzecim członem klasyfikacji utworzonym na podstawie kryterium formy opisu jest podział metod na dwie grupy, a

[3] Działania zawierające niezbędne elementy syntezy i analizy z obu metod można uznać za kompromisowe podejście badawczo-analityczne oparte na tzw. metodzie systemowo-diagnostycznej [23].

[4] Pewne formy morfologii, np. typu AIDA /Analysis of Interconnected Decision Areas/, można uzyskać wykorzystując teorię grafów [12].

[5] Patrz [1,2,9,11,31].

mianowicie: metody opisu słownego i metody opisu formalnego [24].

Metody opisowe polegają na "faktograficznym" odwzorowaniu struktur i procesów zachodzących w analizowanym systemie. W celu wiernego odbicia istniejącej rzeczywistości stosuje się techniki graficznego ujęcia badanych faktów i zdarzeń, a mianowicie: schematy organizacyjne oraz czynnościowe, tablice krzyżowe i decyzyjne, diagramy i sieci działań [3,21].

Charakterystyczną cechą metod opisu formalnego jest dążenie do formułowania modeli matematycznych odwzorowujących teraźniejsze lub przyszłe postacie systemu[6]. Wszystkie zależności i powiązania między elementami systemu są wyrażone za pomocą określonych funkcji obrazujących system nie tylko w stanie statycznym, lecz w jego dynamicznej postaci. W tym ujęciu modele matematyczne stanowią aproksymację rzeczywistości.

Metody opisu formalnego, znajdujące coraz większe zastosowanie, rozwijają się dzięki wykorzystaniu m.in. techniki badań operacyjnych i statystycznych /patrz punkt 3/. Dotychczas omówione metody opisów słownego i formalnego mają określone zalety i wady. Omówiono je w pracy [24].

Ostatnią grupę metod, wyróżnioną ze względu na kryterium wielkości wymaganych nakładów środków /finansowych, czasowych i kadrowych/, stanowią: metody nakładooszczędne i metody nakładochłonne.

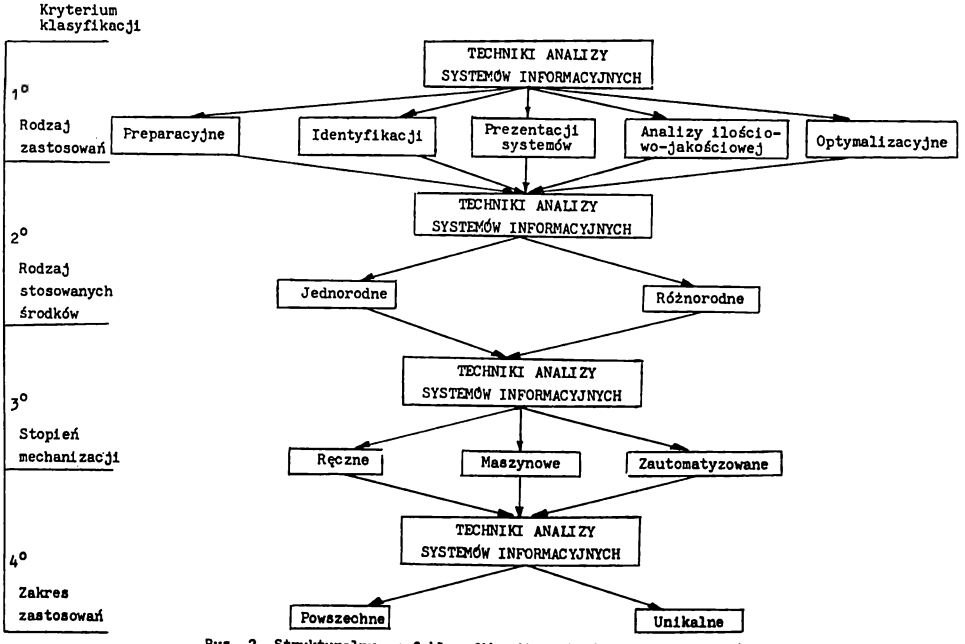
Metody nakładooszczędne wymagają relatywnie małych nakładów środków finansowych, czasowych i kadrowych. Środki techniczne użyte w tych metodach są stosunkowo proste i nieskomplikowane. Najczęściej bowiem stosuje się techniki ręczne lub o małym stopniu mechanizacji /patrz punkt 3/.

Stosowanie metod nakładochłonnych wymaga zaś, w porównaniu z poprzednimi metodami, dużych nakładów finansowych, wiele czasu i licznych zespołów ludzkich. Użyte zaś techniki cechuje wysoki stopień automatyzacji.

3. Klasyfikacja technik analizy

Z podanej na wstępie interpretacji technik wynika, że mają one charakter wykonawczy i służą do realizacji czynności wy-

[6] W tej grupie metod można wyróżnić tzw. modele cybernetyczne, które opisują struktury informacyjne w terminologii cybernetyki.



Rys. 2. Strukturalny graf klasyfikacji technik analizy systemów informacyjnych

nikających z przyjętej metody. O skuteczności stosowania danej metody decyduje nie tylko konsekwentne realizowanie poszczególnych działań badawczo-analitycznych, ale przede wszystkim właściwy dobór technik niezbędnych do realizacji tych działań.

W teorii i praktyce analizy systemów informacyjnych istnieje wiele różnorodnych technik.

Na podstawie szczegółowych badań i uzyskanych doświadczeń empirycznych istniejące techniki można sklasyfikować według grafu podanego na rys. 2. Jak łatwo zauważyć, w grafie tym wyróżniono cztery poziomy, które wynikają z przyjętych kryteriów klasyfikacji.

Pierwszym kryterium klasyfikacji jest rodzaj zastosowań. Na jego podstawie dzieli się techniki na pięć grup, a mianowicie: preparacyjne, identyfikacji, prezentacji systemu, analizy ilościowo-jakościowej oraz optymalizacyjne.

Scharakteryzujemy każdą z nich.

Techniki p r e p a r a c y j n e. Ich celem jest umożliwienie podmiotowi analizującemu przeprowadzenia działań przygotowawczych do ustalenia właściwej strategii analizy konkretnego systemu informacyjnego [22]. Chodzi zatem o sprecyzowanie celu i zakresu analizy, właściwe zaplanowanie realizacji procesu analizy oraz skonkretyzowanie analizowanych problemów badawczych. Tego typu ustalenia można przedstawić za pomocą:

a/ macierzy symetrycznej służącej do oceny zgodności celów analizy z celami obiektu,

b/ wykresów przedstawiających na osi czasu pożądane wartości analizowanych cech systemu,

c/ schematów cybernetycznych ustalających potrzeby użytkownika systemu m.in. w zakresie niezbędnych informacji i kanałów informacyjnych,

d/ metod szacunkowych i porównawczych określających wykonanie poszczególnych czynności analizy,

e/ środków i metod sporządzania planów, jak np. cyklogramy, harmonogramy i metoda PERT,

f/ technik syntezy i analizy umożliwiających łączenie bądź podział zdarzeń i faktów występujących w analizowanym problemie badawczym.

Istotę podgrup technik wyjaśniono w pracach [6,14,16,29].

Techniki i d e n t y f i k a c j i służą do poznania i logicznego powiązania faktów dotyczących systemu i otoczenia, w którym on funkcjonuje [20]. Umożliwiają one przeprowadzenie badań w zakresie:

a/ gromadzenia danych empirycznych za pomocą wywiadów, ankiet, obserwacji i posiedzeń grupowych,

b/ selekcjonowania danych empirycznych za pomocą technik wyboru, kodowania, klasyfikacji i redukcji informacji,

c/ zapisu i utrwalania informacji za pomocą technik ręcznego, zmechanizowanego bądź zautomatyzowanego sposobu dokumentowania zgromadzonego materiału faktograficznego.

Techniki p r e z e n t a c j i s y s t e m ó w. Ta grupa technik jest bardzo liczna i służy do opisu systemu w ujęciu strukturalnym i funkcjonalnym. Badane elementy i relacje można przedstawić za pomocą:

a/ opisu, którego cechą charakterystyczną jest słowne ujęcie istniejących zależności systemu, jego struktury i powiązań wewnętrznych lub zewnętrznych z otoczeniem,

b/ tablic krzyżowych i decyzyjnych,

c/ wykresów, w których wykorzystuje się schematy organizacyjne, czynnościowe, blokowe i sieci działań,

d/ grafów i macierzy.

Obok wymienionych manualnych technik opisu systemów informacyjnych wyróżnia się techniki wspomagane komputerem /patrz trzeci poziom klasyfikacji podanej na rys. 2/.

Szczegółową charakterystykę wyróżnionych technik prezentacji systemu zawiera wiele prac, np. [13,17,21,26].

Techniki a n a l i z y i l o ś c i o w o - w a r t o ś c i o w e j obejmują grupę działań umożliwiających rozpatrywanie elementów systemu nie tylko w aspekcie ilościowym, ale również od strony ich użyteczności w dotychczas funkcjonującym systemie lub w nowo zaprojektowanym. Wykorzystuje się w nich:

a/ narzędzia statystyczne, spośród których na szczególną uwagę zasługują parametry opisowe, rozkład normalny, regresja i korelacja, weryfikacja hipotez oraz ekstrapolacja szeregów czasowych [4,28],

b/ badania porównawcze, takie jak technika delficka i badanie Shang [21].

Techniki o p t y m a l i z a c y j n e umożliwiają wybór alternatywnych rozwiązań analizowanych problemów. Do najczęściej

stosowanych technik tej grupy należą:

a/ badania operacyjne, np. programowanie liniowe, programowanie wielokryteriowe, technika powiązań sieciowych, teoria kolejek i teoria gier [7,18,27],

b/ symulacja matematyczna, którą stosuje się do obserwacji zachowania systemu w różnych warunkach, tworzących określone sytuacje problemowe [11].

Kolejnym kryterium podziału technik analizy jest rodzaj stosowanych środków. Dzięki niemu można dzielić zbiór technik na techniki jednorodne i techniki różnorodne. W technikach j e d n o r o d n y c h wykorzystuje się jeden lub kilka zbliżonych rodzajów środków. Ich stosowanie nie wymaga wszechstronnego przygotowania zawodowego podmiotu analizującego. Wystarcza przygotowanie specjalistyczne jednokierunkowe, np. w zakresie technik ręcznych, maszynowych czy zautomatyzowanych. Inaczej rzecz się ma w odniesieniu do technik r ó ż n o r o d n y c h. W procesach analizy korzysta się zwykle z wielu znacznie zróżnicowanych technik. Ich stosowanie wymaga odpowiedniego, wszechstronnego przygotowania zawodowego podmiotu analizującego lub wymusza stworzenie mieszanego zespołu analityków specjalistów znających określone typy technik analizy.

Następnym kryterium podziału jest stopień mechanizacji. Z tego względu ogół technik dzielimy na trzy grupy, a mianowicie: ręczne, maszynowe i zautomatyzowane.

Techniki r ę c z n e. Do tej grupy należą tradycyjnie techniki służące do identyfikacji i opisu systemu /patrz pierwszy poziom klasyfikacji na rys. 2/. Wykorzystuje się tu środki najprostsze i nieskomplikowane w działaniu. Najczęściej wspomagane są przez środki orgatechniczne. Dzięki prostocie stosowano je w analizie od dawna, dlatego mają bogatą tradycję. Jednakże na obecnym poziomie rozwoju informatyki są stopniowo wypierane przez techniki maszynowe i zautomatyzowane.

Techniki m a s z y n o w e wykorzystują dostępne urządzenia mechaniczne, np. arytmometr, kalkulatory, maszyny księgujące, maszyny fakturujące, maszyny księgująco-fakturujące. Stosuje się je głównie do wykonywania prostych, powtarzalnych działań arytmetycznych i sporządzania elementarnych zestawień wynikowych [8].

Techniki z a u t o m a t y z o w a n e są pod względem technicznym najbardziej złożone. Stosowanie ich wymaga specjali-

stycznego przygotowania zawodowego podmiotu analizującego. Ostatnio coraz częściej korzysta się z technik lansujących analizę wspomaganą komputerem. Wiele firm światowych opracowało i wdrożyło odpowiednie produkty programowe. Do najbardziej popularnych technik analizy można zaliczyć[7]:

a/ technikę ADS /Accuratelly Defined System/ firmy NCR, gdzie podstawą opisu systemu są formularze wejścia, wyjścia, procedur przetwarzania oraz tablice decyzyjne,

b/ technikę SOP /Study Organization Plan/ firmy IBM, która umożliwia analizę zasobów, czynności i przepływu informacji na odpowiednio zaprojektowanych tablicach i schematach,

c/ technikę TAG /Time-Automated Giret/ powstałą w IBM. Podobnie jak w poprzednich technikach tak i w tej najpierw analizuje się koszty informacyjne systemu, po czym projektuje się wejścia na podstawie uprzednio sformułowanych wyjść systemu informatycznego,

d/ technikę PSL/PSA bazującą na technice ADS i TAG, opracowaną przez zespół badawczo-naukowy Uniwersytetu Michigan. Część podstawowa tej techniki dotyczy tzw. języka opisu PSL /Problem Statement Language/, który analizuje parametry wejścia systemu i warunki powstania wyjścia systemu informacyjnego. Część druga, tzw. PSA /Problem Statement Analyzer/, służy diagnostyce sieci informacyjnej. Technika ta jest przydatna zwłaszcza do aktualizowania dokumentacji opisu systemu[8].

Ostatnim kryterium proponowanej klasyfikacji technik analizy jest zakres ich stosowania. Za pomocą tego kryterium dzielimy techniki na powszechnie stosowane oraz unikalne.

Techniki p o w s z e c h n i e s t o s o w a n e są ogólnie dostępne i wykorzystywane w analizach przez większość ośrodków informatyki na świecie. Przykładami takich technik są schematy blokowe, tablice decyzyjne i sieci czynności.

Techniki u n i k a l n e zazwyczaj opracowuje się i stosuje w szczególnie złożonych i niepowtarzalnych pracach badawczo-analitycznych. Na ogół nie są to techniki powszechnie znane, dostępne, a niekiedy wręcz stanowią tajemnicę firm prowadzących prace z dziedziny informatyki stosowanej.

---

[7] Obszerny przegląd tego typu technik i wykaz literatury na ten temat zawiera praca [26].

[8] Pakiet ten posiada Ośrodek Komputerowy AE we Wrocławiu i służy podczas nauczania analizy systemów informacyjnych.

LITERATURA

[1] R.L. A c k o f f, M.W. S a s i e n i, Fundamentals of Operations Research, New York 1968 /tłum. ros. Osnovy issledovanija operacii, Moskva 1971, Mir/.

[2] W.A. B o c c h i n o, Systemy informacyjne zarządzania. Narzędzia i metody, Warszawa 1975, WNT.

[3] A. D a n i e l s, D. Y a t e s, Podstawy analizy systemów, Warszawa 1974, WNT.

[4] M. D o b i j a, E. S m a g a, B. W ą s i k, K. Z a j ą c, Metody statystyczne w systemowym przetwarzaniu informacji ekonomicznej, Warszawa 1983, PWE.

[5] J.L. F r ą c k i e w i c z, Systemy sprawnego działania. Wrocław-Warszawa-Kraków-Gdańsk 1980, Ossolineum.

[6] I. H a b r, I. V e p ř e k, Systemowa analiza i synteza, Warszawa 1976, PWE.

[7] E. I g n a s i a k, Teoria grafów i planowanie sieciowe, Warszawa 1976, PWE.

[8] Informatyka. Wstęp do informatyki. Technologia przetwarzania danych, red. E. Niedzielska, Warszawa 1983, PWE.

[9] A. K a u f m a n, A. F a u r e, Badania operacyjne na co dzień. Warszawa 1968, PWN.

[10] J. K i s i e l n i c k i, Metody informatyczne, Warszawa 1981, PWE.

[11] L. K o n d r a t o w i c z, Modele symulacyjne systemów, Warszawa 1978, WNT.

[12] B. K o r z a n, Elementy teorii grafów i sieci. Metody i zastosowania, Warszawa 1978, WNT.

[13] B. L a n g e f o r s, Theoretical Analysis of Information Systems. Studentenliteratur, Lund 1966.

[14] B.W. L i n d g r e n, Elementy teorii decyzji, Warszawa 1977, WNT.

[15] A. M a ł a c h o w s k i, Przedmiot badań informatyki, w: Zarys informatyki dla ekonomistów, red. A. Nowicki, Akademia Ekonomiczna we Wrocławiu, /w druku/.

[16] Z. M i k o ł a j c z y k, Metody organizowania pracy w warunkach nowoczesnego przemysłu, Warszawa 1973, PWE.

[17] A.A. M o d i n, I.S. Z i n g e r, M.F. K o r o t a e v, Issledovanie i analiz potokov informacii na promyšlennych predprijatijach, Moskva 1970, Nauka.

[18] Y. M u l l e r, Wprowadzenie do nauki organizacji i badań operacyjnych, Warszawa 1971, PWE.

[19] E. N i e d z i e l s k a, Funkcjonowanie i rozwój informatycznych systemów zarządzania. Koncepcja. Teoria. Praktyka. Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 113, Wrocław 1977.

[20] A. N o w i c k i, Identyfikacja i analiza systemu przetwarzania informacji, [w:] Automatyczne przetwarzanie informacji, red. Z. Hellwig, Warszawa 1982, PWE.

[21] A. N o w i c k i, Modernizacja systemu informacyjnego w przedsiębiorstwie przemysłowym, Warszawa 1979, PWE.

[22] A. N o w i c k i, Przesłanki i strategia analizy, [w:] Organizacja przetwarzania danych, cz. I – Narzędzia. Metody, red. A. Nowicki, Akademia Ekonomiczna, Wrocław 1981.

[23] A. N o w i c k i, Systemowo-diagnostyczna metoda badania i analizy systemów informacyjnych obiektów gospodarczych. [w:] Informatyka. Teoria. Praktyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 235, Wrocław 1983.

[24] A. N o w i c k i, Teoria i praktyka analizy systemów informacyjnych, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 151, Wrocław 1979.

[25] Projektowanie systemów informatycznych, cz. I - Metody projektowania systemów informatycznych, red. W. Olejniczak i T. Wierzbicki, Politechnika Szczecińska, Szczecin 1983.

[26] Projektowanie systemów informatycznych wspomagane komputerem, cz. I – Analiza możliwości. Praca zbiorowa pod kierunkiem naukowym J. Wojdyły, Akademia Ekonomiczna we Wrocławiu, 1980 /maszynopis/.

[27] W. R a d z i k o w s k i, Metody matematyczne i statystyczne w przedsiębiorstwie, Warszawa 1976, PWE.

[28] J. S t e c z k o w s k i, A. Z e l i a ś, Statystyczne metody analizy cech jakościowych, Warszawa 1981, PWE.

[29] L.W. S z y b i ń s k i, Projektowanie organizacji przedsiębiorstw, Warszawa 1979, PWE.

[30] I. T r z c i e n i e c k i, Projektowanie systemów zarządzania, Warszawa 1980, PWN.

[31] H.M. W a g n e r, Badania operacyjne w zarządzaniu, Warszawa 1980, PWN.

[32] J. Z i e l e n i e w s k i, Organizacja i zarządzanie, Warszawa 1969, PWN.

AN ATTEMPT TO CLASSIFY THE METHODS AND TECHNIQUES OF INFORMATION SYSTEMS ANALYSIS

Summary

In the article an attempt is made to systematically discuss of the methods and techniques of informatic systems analysis. Appropriate division criteria, referring to the so-called integrating classification, are formulated. The resulting subfiles of the methods and techniques of such an analysis are then characterized by their basic features and properties.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Wojciech D o m i ń s k i
Akademia Ekonomiczna we Wrocławiu

PRÓBA KOMPLEKSOWEGO UJĘCIA STRUKTURY EKSPLOATACJI INFORMATYCZNEGO SYSTEMU ZARZĄDZANIA

1. Wprowadzenie

Dotychczas w teorii i praktyce informatycznych systemów zarządzania niewiele miejsca poświęcono problemowi bieżącej eksploatacji systemów, badania bowiem skupiono w zasadzie na zagadnieniach tworzenia informatycznych systemów zarządzania, a zatem ich projektowania, oprogramowania i wdrażania. Doskonalono metody i techniki mające ułatwić i przyspieszyć tworzenie systemu oraz podwyższyć jakość uzyskiwanych rozwiązań. Należy zwrócić uwagę, że podejmowane działania były w mniejszym lub większym stopniu skoncentrowane /producent sprzętu i oprogramowania, wszelkiego rodzaju instytucje naukowo-badawcze, wyższe uczelnie/, co sprzyjało prowadzeniu odpowiednich badań.

Problemy bieżącej eksploatacji wdrożonych systemów, w odróżnieniu od zagadnień tworzenia informatycznych systemów zarządzania są rozwiązywane niejako doraźnie przez samych użytkowników. Naturalne rozproszenie użytkowników utrudnia zarówno wymianę uzyskiwanych doświadczeń, jak i rozwój teoretycznych podstaw eksploatacji informatycznych systemów zarządzania.

Można sądzić, że w kraju w najbliższym okresie informatyka może się rozwijać dzięki doskonaleniu już eksploatowanych systemów i podwyższaniu ich efektywności. Nakazem chwili staje się zdecydowane przejście od ekstensywnej informatyzacji zarządzania do informatyzacji intensywnej, której celem jest lepsze wykorzystanie zatrudnianych kadr oraz posiadanych zasobów technicznych i programowych. Z tych względów wzrasta rola badań i prac mających usprawnić bieżącą eksploatację systemów oraz zwiększyć

jej efektywność /zarówno przez obniżkę kosztów, jak i przede wszystkim powiększenie efektów/[1].

Celem artykułu jest próba kompleksowego ujęcia struktury eksploatacji informatycznego systemu zarządzania. Dotychczasowe badania tego zagadnienia miały - zdaniem autora - charakter wycinkowy, co uniemożliwiło stworzenie teorii eksploatacji systemów tej klasy[2]. Ponieważ w literaturze przedmiotu nie przedstawiono takiego podejścia, zatem wydaje się, że próba ta może zainicjować systematyczne badania w tym zakresie. Ostatecznym celem tych badań powinno być opracowanie metodologii tworzenia racjonalnych systemów sterowania eksploatacją informatycznych systemów zarządzania.

2. Struktura eksploatacji informatycznego systemu zarządzania

Przez e k s p l o a t a c j ę informatycznego systemu zarządzania należy rozumieć całokształt działań technicznych, organizacyjnych itp. w zakresie planowania, kontroli i realizowania, podejmowanych po to, aby użytkować oraz obsługiwać system zgodnie z celem, do którego został stworzony[3].

U ż y t k o w a n i e to korzystanie z systemu zgodnie z jego przeznaczeniem, którym jest uzyskiwanie określonych informacji w ściśle określonym czasie.

O b s ł u g i w a n i e systemu polega na odnawianiu jego walorów użytkowych w celu przywrócenia, podtrzymania lub rozwoju jego pełnej zdolności do realizacji zadań użytkowych.

Eksploatację informatycznego systemu zarządzania można rozpatrywać dwojako. Zgodnie z podejściem f u n k c j o n a l n y m, eksploatacja to jednoczesne realizowanie procesów użytkowania i obsługiwania systemu. Procesy te stanowią określone ciągi nastę-

---

[1] Koszty eksploatacji informatycznych systemów zarządzania zależą głównie od rozwiązań przyjętych w fazie tworzenia systemu.

[2] Należy podkreślić, że w zakresie teorii eksploatacji systematycznie bada się jedynie eksploatację urządzeń technicznych. Przykład stanowi teoria eksploatacji urządzeń, opracowana przez J. Koniecznego /por. [2; 3]/. Za celowe należy uznać wykorzystanie metodologicznych osiągnięć i wielu pojęć tej teorii na potrzeby eksploatacji informatycznych systemów zarządzania.

[3] Por. [2; 4].

pujących po sobie i wzajemnie uzależnionych czynności i zdarzeń prowadzących do założonych celów.

Wymienione procesy są dość skomplikowane, gdyż przedmiotem eksploatacji nie jest pojedyncze urządzenie czy grupa urządzeń, lecz informatyczny system zarządzania będący układem o szczególnej złożoności. Kompleksową analizę eksploatacji systemu w aspekcie funkcjonalnym /umożliwiającą poznanie mechanizmów determinujących funkcje i przebieg procesów użytkowania i obsługiwania systemu/ powinno poprzedzić jej s t r u k t u r a l n e ujęcie. Podejście to umożliwi rozwinięcie pojęć użytkowania i obsługiwania systemu informatycznego w aspekcie jego struktury oraz relacji wewnątrzsystemowych.

Zgodnie z ujęciem strukturalnym informatyczny system zarządzania stanowi "wyróżniony przestrzennie i uporządkowany czasowo zbiór podsystemów informacji, jej nadawców i odbiorców oraz reguł i środków transformacji"[4]. Wymienione podsystemy stanowią zatem przedmiot eksploatacji, a więc są zarówno użytkowane, jak i obsługiwane.

Na potrzeby dalszych rozważań wyróżnione podsystemy można poddać pewnemu scaleniu. Posłuży do tego najprostsza, pragmatyczna definicja systemu informatycznego, stanowiąca, że jest on skomputeryzowanym systemem informacyjnym. Zatem struktura systemu informatycznego obejmuje system informacyjny oraz zbiór reguł i środków technicznych, umożliwiających automatyczne przetwarzanie informacji. Wymieniony zbiór reguł, zaczerpnięty z wiedzy o metodach, procesach, sposobach i środkach transformacji, stanowi technologię automatycznego przetwarzania informacji, a środki techniczne odpowiadają pojęciu systemu komputerowego.

Za przedmiot eksploatacji można więc uznać trzy sprzężone elementy, którymi są:

- system informacyjny,
- system komputerowy, rozumiany jako zorganizowany zbiór środków technicznych informatyki, oprogramowania i obsługi /operatorskiej i konserwatorskiej/[5],
- technologia automatycznego przetwarzania informacji.

Ostatni z wymienionych elementów stanowi spoiwo łączące dwa poprzednie elementy w jeden system.

---

[4] Patrz [5, s. 29].

[5] Por. [1].

Eksploatacja systemu informatycznego obejmuje zatem użytkowanie i obsługiwanie systemu informacyjnego, systemu komputerowego oraz technologii automatycznego przetwarzania informacji. Zatem przyjęty w teorii zakres pojęcia eksploatacji informatycznego systemu zarządzania jest znacznie większy niż w praktyce[6].

3. Elementarny układ eksploatacji informatycznego systemu zarządzania

W celu przedstawienia elementarnego układu eksploatacji systemu informatycznego można posłużyć się modelem prakseologicznym uwypuklającym pojęcie łańcucha realizowanych działań.

Ł a ń c u c h e m d z i a ł a n i a są trzy uporządkowane elementy, z których pierwszy spełnia rolę podmiotu działania, drugi - rolę pośrednika, tj. narzędzia pośredniczącego w działaniu, trzeci zaś - rolę przedmiotu, tj. tworzywa, będącego przedmiotem działania[7].

W kontekście podanej definicji u ż y t k o w a n i e systemu informatycznego będzie stanowić łańcuch działań, w którym rolę podmiotu spełnia użytkownik systemu, rolę pośrednika działania - system informatyczny, a rolę przedmiotu - informacje. Analogicznie, o b s ł u g i w a n i e m systemu informatycznego nazwiemy łańcuch działań, w którym podmiotem jest obsługa[8] systemu, pośrednikiem - zespół metod, technik i narzędzi, stosowanych w konserwacji i modernizacji systemu informatycznego, oraz ludzie posługujący się tymi środkami /wymienione elementy nazwiemy środkami obsługi/, przedmiotem - system informatyczny lub jego elementy. Zaproponowane ujęcie eksploatacji systemu informatycznego przedstawia rys. 1.

Schemat użytkowania i obsługiwania informatycznego systemu zarządzania jest bardzo uproszczony. Aby go urealnić należy uwzględnić przedstawioną wyżej strukturę systemu informatycznego oraz wnioski płynące z relacji istniejących między systemem informatycznym, systemem informacyjnym oraz systemem zarządzania[9].

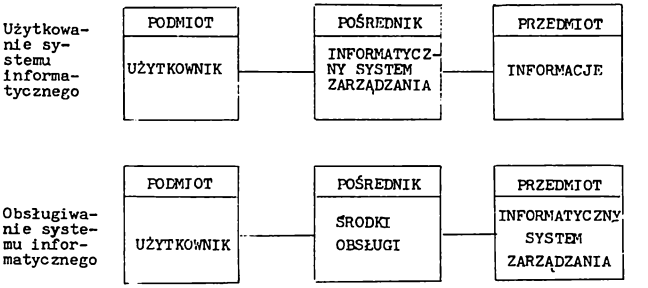
---

[6] Rozmiary artykułu nie pozwalają na bardzo dokładne omówienie problemów dotyczących użytkowania i obsługiwania elementów systemu informatycznego.

[7] Patrz [2, s. 33].

[8] Termin "obsługa" zarezerwowany jest dla podmiotu procesu obsługiwania systemu informatycznego.

[9] Problematykę relacji międzysystemowych wyczerpująco przedstawiono w [5, s. 31-36].



Rys. 1. Użytkowanie i obsługiwanie informatycznego systemu zarządzania

Uwzględnienie struktury systemu pozwala na przedstawienie eksploatacji informatycznego systemu zarządzania jako trzech równoległych strumieni działania, dotyczących procesów użytkowania i obsługiwania systemu informacyjnego /SI/, systemu komputerowego /SK/ oraz technologii automatycznego przetwarzania informacji /T/ /por. tab. 1/.

Tabela 1

Użytkowanie i obsługiwanie informatycznego systemu zarządzania

Element systemu informatycznego	Stan eksploatacji	Podmiot	Pośrednik	Przedmiot
SI	Użytkowanie	Użytkownik SI	SI	Informacje
	Obsługiwanie	Obsługa SI	Środki obsługi SI	SI
SK	Użytkowanie	Użytkownik SK	SK	Informacje
	Obsługiwanie	Obsługa SK	Środki obsługi SK	SK
T	Użytkowanie	Użytkownik T	T	Informacje
	Obsługiwanie	Obsługa T	Środki obsługi T	T

Źródło: Opracowanie własne.

Podmioty procesów użytkowania i obsługiwania elementów systemu informatycznego określone w tabeli 1, budzą pewne wątpliwości. Podmiotami tymi w każdym przypadku są ludzie inicjujący określony łańcuch działań. W praktyce bowiem nie ma ścisłego podziału kadry na użytkowników i obsługę poszczególnych elementów systemu. Bardzo często ci sami pracownicy użytkują i obsługują określony element systemu bądź też użytkują lub/i obsługują kilka elementów systemu.

Należy jednak zwrócić uwagę na fakt, że podmiotami działania są najczęściej ludzie stanowiący integralną część systemu informatycznego[10]. Można więc stwierdzić, że bezpośredni użytkownicy i bezpośrednia obsługa znajdują się "wewnątrz systemu", stanowią jego elementy składowe.

Uwzględniając z kolei relacje międzysystemowe w obiekcie mikroekonomicznym należy przypomnieć, że system informatyczny stanowi układ ściśle podporządkowany nadrzędnemu systemowi informacyjnemu, który pełni służebną rolę w stosunku do systemu zarządzania.

Rozważając problematykę struktury eksploatacji informatycznego systemu zarządzania w relacjach międzysystemowych można pominąć udział systemu informacyjnego. System informatyczny jest skomputeryzowaną c z ę ś c i ą informacyjnego systemu zarządzania. W praktyce bowiem nie zdarza się, aby zakresy obu systemów dokładnie się pokrywały. Wynika z tego, że w naszych rozważaniach związki między systemem informacyjnym a informatycznym można uważać za nieistotne[11]. Związki te tkwią bowiem w systemie informatycznym i stanowią jego integralną część.

W tym kontekście ważne stają się wzajemne powiązania między systemem informatycznym a systemem zarządzania, rozumianym jako "ogół metod i technik systemowego sterowania transformacją zasileń oraz /lub/ informacji w układzie gospodarczym"[12]. System zarządzania określa więc zarówno cel, jak i zakres działania systemu informatycznego.

---

[10] Wyjątek stanowi obca kadra, świadcząca usługi z zewnątrz systemu związane z serwisem technicznym i oprogramowaniem, szkoleniem kadry użytkownika, ekspertyzami, analizami, pracami adaptacyjnymi i modernizacyjnymi.

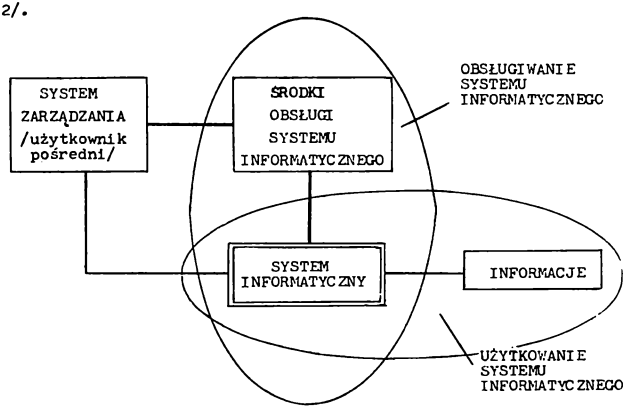
[11] Oczywiście tych relacji nie można pominąć w problematyce tworzenia systemu informatycznego.

[12] Patrz [5, s. 23].

W u k ł a d z i e u ż y t k o w a n i a systemu informatycznego należy uwzględnić nie tylko strukturę systemu oraz relacje międzysystemowe, lecz także uogólnionego, pośredniego użytkownika systemu, którym jest system zarządzania, oraz użytkowników bezpośrednich, będących podmiotami łańcucha użytkowania, tj. ludzi bezpośrednio zaangażowanych w przetwarzanie danych, którzy są elementem systemu informatycznego i jednocześnie jego użytkownikami.

Analogicznie, w u k ł a d z i e o b s ł u g i w a n i a systemu informatycznego podmiotem bezpośrednim jest obsługa bezpośrednia systemu, stanowiąca jego element, podmiotem pośrednim również system zarządzania. Można tak uważać, gdyż system zarządzania inicjuje w obrębie systemu informatycznego wszelkie działania związane z konserwacją, modernizacją lub rozwojem, które same w sobie stanowią procesy obsługiwania systemu.

W wyniku syntezy dotychczasowych rozważań można przedstawić elementarny układ eksploatacji systemu informatycznego /por. rys. 2/.



Rys. 2. Elementarny układ eksploatacji informatycznego systemu zarządzania

Takie ujęcie problemu staje się podstawą analizy elementarnych układów eksploatacji dla składowych systemu informatycznego, tj. systemów informacyjnego i komputerowego oraz technologii

przetwarzania. Ponieważ każda z wymienionych składowych ma odrębną strukturę, przeto w dalszej analizie można uwzględniać elementy cząstkowych struktur systemu.

4. Wnioski końcowe

Zaproponowane ujęcie struktury eksploatacji informatycznego systemu zarządzania można uznać za wstęp do dalszych badań w tym zakresie. Wydaje się bowiem, że jedynie systemowa analiza układów eksploatacji poszczególnych elementów strukturalnych systemu informatycznego pozwoli na określenie tych związków i relacji wewnątrzsystemowych oraz oddziaływań z zewnątrz, które decydują o sprawności i efektywności procesu eksploatacji, tj. zarówno procesów użytkowania, jak i obsługiwania systemu. Aby jednak taka analiza przyniosła spodziewane efekty, musi być wsparta analizą funkcjonowania poszczególnych ogniw systemu eksploatacji.

Wydaje się, że zaproponowane podejście ułatwi prowadzenie analizy systemu kierowania eksploatacją, a w dalszym ciągu - projektowania takiego systemu. Analiza podmiotów wszelkich działań w obrębie procesu eksploatacji umożliwi bowiem wyodrębnienie szczegółowych relacji między poszczególnymi podmiotami i tym samym da podstawy do określenia stopnia ich wzajemnej zależności, zakresu kompetencji itp.

Należy sądzić, że wykorzystanie strukturalnego ujęcia problemu eksploatacji stanowi punkt wyjścia do opracowania modeli różnych wariantów eksploatacji informatycznych systemów zarządzania.

LITERATURA

[1] W. D o m i ń s k i, Techniczne i ekonomiczne przesłanki doboru sprzętu komputerowego, [w:] Ekonomika systemów informatycznych, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 98, Wrocław 1977.

[2] J. K o n i e c z n y, Sterowanie eksploatacją urządzeń. Warszawa 1975, PWN.

[3] J. K o n i e c z n y, Wstęp do teorii eksploatacji urządzeń. Warszawa 1971, WNT.

[4] Mały słownik cybernetyczny, red. M. Kempisty, Warszawa 1973, WP.

[5] E. N i e d z i e l s k a, Funkcjonowanie i rozwój informatycznych systemów zarządzania. Koncepcja. Teoria. Praktyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 113, Wrocław 1977.

AN ATTEMPTED COMPREHENSIVE APPROACH TO THE STRUCTURE OF COMPUTER MANAGERIAL SYSTEM EXPLOITATION

Summary

The article presents a comprehensive approach to the structure of informatic managerial system exploitation, thus being a valuable contribution to future research of system exploitation and the growth of its effectiveness. Three subsystems have been distinguished in the structure of an informatic system. These are an information system, a computer system, and an automatic data processing system.

An elementary scheme of an informatic system exploitation, involving its structure, utilization, and attendance objects as well as the relation between informatic and managerial systems, is also presented in the paper.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Jerzy K o r c z a k
Akademia Ekonomiczna we Wrocławiu

PROJEKTOWANIE DIALOGU W KOMPUTEROWYCH SYSTEMACH UCZĄCYCH /Na przykładzie systemu OSKAR/

1. Wprowadzenie

W artykule podzielimy się doświadczeniami zdobytymi w czasie czterech lat projektowania dialogów do systemu nauczania wspomaganego komputerem OSKAR[1]. Wiedzę na temat projektowania materiału dydaktycznego dla systemu zaczerpnęliśmy z fundamentalnej pracy J. Martina "Dialog człowieka z maszyną cyfrową". Skorzystaliśmy też z doświadczeń i materiałów publikowanych przez czołowe ośrodki zajmujące się problematyką komputeryzacji nauczania, w tym głównie zespoły D. Bitzera [1,11], A. Borka [2-4] i H.O'Neila [10]. Ponieważ w naszym kraju nauczanie z zastosowaniem komputerów jest w fazie eksperymentu, zasady i wskazówki dla projektantów dialogów uczących mają charakter częściowo subiektywny i częściowo uwarunkowany możliwościami sprzętowo-programowymi[2].

[1] System OSKAR opracowany w Akademii Ekonomicznej we Wrocławiu ma służyć indywidualnemu nauczaniu, tzn takiemu, w którym zakres, szczegółowość, tempo i intensywność nauki są dostosowane do predyspozycji i potrzeb studenta. Nauka odbywa się w czasie dialogu studenta z systemem, przy czym w tym samym czasie system może równocześnie nauczać różnych przedmiotów wielu studentów. Bardziej szczegółowe informacje na temat konstrukcji i funkcjonowania systemu OSKAR można znaleźć w publikacjach [6,8].

[2] Proces nauczania jest realizowany przy wykorzystaniu systemu komputerowego R-32, pracującego pod kontrolą systemu operacyjnego OS/JS MVT z opcją TSO. Dialog studenta z systemem prowadzony jest na monitorze ekranowym EC 7917.

W systemie wyróżniono osiem trybów pracy[3], realizujących główne zadania procesu nauczania. Każdy tryb może funkcjonować samodzielnie lub w połączeniu z innym. Podstawowym trybem pracy OSKAR jest tryb NAUKA, którego zadaniem jest udostępnianie studentom materiału dydaktycznego, nauczanie, sprawdzanie postępów w nauce oraz gromadzenie danych statystycznych o pracy studenta. Student pracujący w systemie może, nie przerywając nauki, korzystać z ułatwień dostępnych w trybach POMOC, OBLICZENIA i PROGRAMOWANIE. Schemat dialogu studenta z systemem OSKAR podczas procesu nauczania przedstawia rys. 1. Pokazano na nim, że zasadniczy dialog studenta z systemem sterowany jest przez interaktywny program uczący zapisany w języku autorskim PASCAL-A.

2. Struktura materiału dydaktycznego

Z punktu widzenia struktury tematycznej nauczanego przedmiotu treść zawarta w programie komputerowym stanowi fragment materiału nazywanego jednostką dydaktyczną[4]. W każdej jednostce dydaktycznej można wyróżnić pięć głównych części. Oto one:

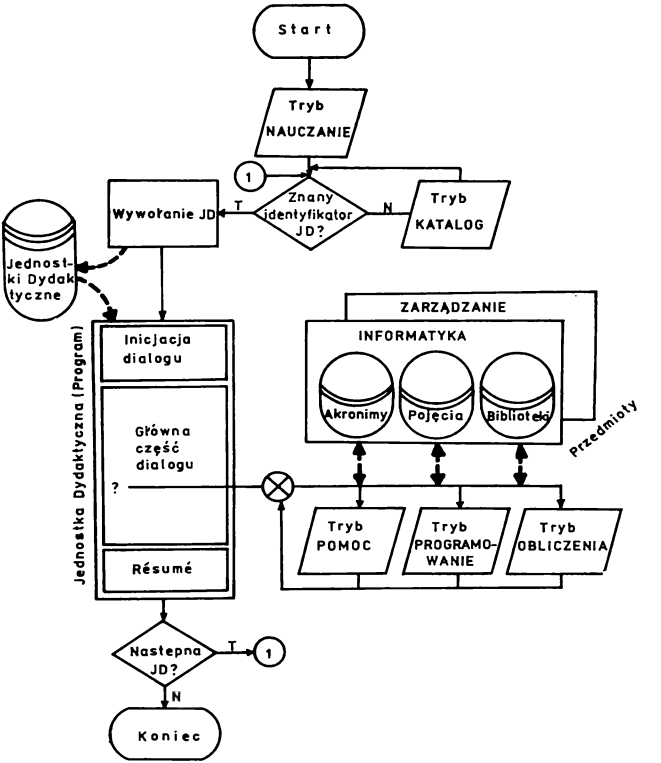
- strona tytułowa zawierająca tytuł, nazwiska autorów, miejsce i czas wdrożenia,
- wprowadzenie do jednostki opisujące cel nauczania, tematykę, wymagania, zasady prowadzenia dialogu, przecietnv czas trwania dialogu,
- część główna jednostki, zawierająca zasadniczy materiał dydaktyczny,
- część testująca /opcyjnie/,
- zakończenie jednostki podające ocenę pracy studenta, spis literatury, zalecenia itp.

W projektowaniu konstrukcji jednostki stosuje się trzy podstawowe struktury: liniową, hierarchiczną i sieciową[5]. Struktura

[3] Trybem pracy systemu OSKAR nazywa się zestaw tematycznie i funkcjonalnie powiązanych programów i zbiorów danych służących do realizacji jednego z wyróżnionych zadań systemu /patrz [8]. Należy zaznaczyć, że pojęcie to różni się od przyjętego w informatyce terminu trybu pracy, oznaczającego pewien reżim technologiczny realizacji zadania.

[4] Przyjęcie takiej nazwy było podyktowane koniecznością zaznaczenia odmienności formy i zakresu tematycznego jednostki dydaktycznej od powszechnie przyjętego w nauczaniu tradycyjnym podziału na lekcje, wykłady, ćwiczenia itp. Patrz [6].

[5] Patrz [7].



Rys. 1. Ogólny schemat procesu nauczania w systemie OSKAR

liniowa jest najłatwiejsza w projektowaniu i prosta w sterowaniu procesem. Tylko niektóre jednostki mają strukturę liniową, np. jednostki informacyjne o systemie OSKAR. Większość jednostek ma natomiast strukturę hierarchiczną /indeksową/ i sieciową. Walorem struktury hierarchicznej jest duża klarowność podziału treści materiału na tematy, części czy zagadnienia. Elementy jednostki są zapisane w postaci niezależnych modułów /zestawu procedur/, które student wybiera sam stosownie do opanowania tematu. Równie często stosuje się strukturę sieciową. Pozwala ona na indywidualizowanie procesu nauczania w zależności od postępów w nauce studenta, wyników testów i posiadanych przez niego wiadomości. Wybór treści należy tutaj też do studenta, ale jest w pewnym stopniu ograniczony z racji zdefiniowanych przez autora dopuszczalnych dróg poznania materiału dydaktycznego. W systemie OSKAR większość jednostek ma mieszaną strukturę oraz stosuje zróżnicowane metody i techniki sterowania dialogiem. Na przykład część główna programu może mieć strukturę hierarchiczną, a poszczególne moduły strukturę sieciową.

W sensie technologii przetwarzania jednostka dydaktyczna jest - jak powiedzieliśmy - programem interaktywnym. Czas realizacji programu zależy głównie od indywidualnych postępów w nauce studenta. W systemie OSKAR przyjęliśmy, że czas realizacji jednostki dydaktycznej nie powinien przekraczać 45 minut. Założenie to wynikało z przyjętych w szkolnictwie zasad organizacji zajęć oraz możliwości koncentracji i efektywnej współpracy studenta z systemem. Z przeglądu i analizy kilkunastu programów wynika, że przeciętny czas nauki w jednostce w większości tematów nie trwa dłużej niż 30 minut. Obejmuje zatem fizycznie od 8 do 10 ekranów dialogu. Każdy z ekranów może zawierać treść informacyjną, przykład, ćwiczenie, test itp. Nasze skromne doświadczenia z eksploatacji nie pozwalają na ogólną ocenę przyjętego rozwiązania.

3. Rodzaje dialogów w programach dydaktycznych

Jednym z zasadniczych czynników decydujących o efektywności nauczania w systemach komputerowych /a zatem i czasie realizacji materiału/ jest sposób rozwiązania dialogu studenta z systemem.

Przez dialog rozumie się "konwersację" studenta z nauczycielem za pośrednictwem systemu komputerowego.

Pierwszym problemem, z którym styka się początkujący projektant dialogu, jest wybór tematyki materiału dydaktycznego. W literaturze[6] można wprawdzie znaleźć pewne ogólne charakterystyki dobrze zaprojektowanego dialogu, ale brakuje sugestii i porad, jak podejść do rozwiązania tego problemu. Powszechnie wiadomo, że przepisanie fragmentów treści książki czy skryptu /nawet napisanego w konwencji programowanego nauczania/ na ekran monitora jest stratą czasu i kosztów. Jedną z metod wyboru treści dialogu jest analiza tematyki przedmiotu ze szczególnym uwzględnieniem tych tematów, w których zastosowanie komputera zapewniłoby podniesienie efektywności nauczania. Chodzi tu o wskazanie wszystkich tematów, których opanowanie przez uczniów zależy od ich pojemności pamięci, szybkości działania i wykonywania obliczeń, podzielności uwagi, cierpliwości itp. Wyszczególnienie tych "wąskich gardeł" procesu dydaktycznego jest niezmiernie trudne i wymaga bardzo dobrej znajomości przedmiotu oraz dużego doświadczenia pedagogicznego. Zalecane jest zatem przedyskutowanie w gronie specjalistów zakresu tematycznego przedmiotu i nowych zadań dydaktycznych wynikających z możliwości wprowadzenia komputera. Należy też rozważyć wprowadzenie tematów, które w poprzedniej technice nauczania uważano za bardzo trudne.

Metodę tę zweryfikowaliśmy w projektowaniu dialogów do materiału dydaktycznego z zakresu dwóch przedmiotów: "Analizy systemów informacyjnych" i "Projektowania systemów informatycznych"[7]. Zaznaczmy, że jest ona szczególnie efektywna, gdy zespół autorów nie ma jeszcze doświadczenia w projektowaniu dialogów student z komputerem.

W projekcie nowej jednostki dydaktycznej należy uwzględnić zgodność celów jednostki nie tylko z celami dydaktycznymi przedmiotu, ale także z pożądanym profilem wiedzy studenta. Dlatego student powinien opanować materiał, którego znajomość będzie potrzebna w jego przyszłej pracy, pomocna w rozwijaniu zdolności kreowania i samodoskonalenia. Nieuwzględnienie tej idei, przejawiające się w koncentracji materiału tylko na aktualnych proble-

[6] Patrz [4;9;11].

[7] Patrz 5 .

mach nauczania, prowadzi zwykle do szybkiego "starzenia się" treści materiału dydaktycznego[8].

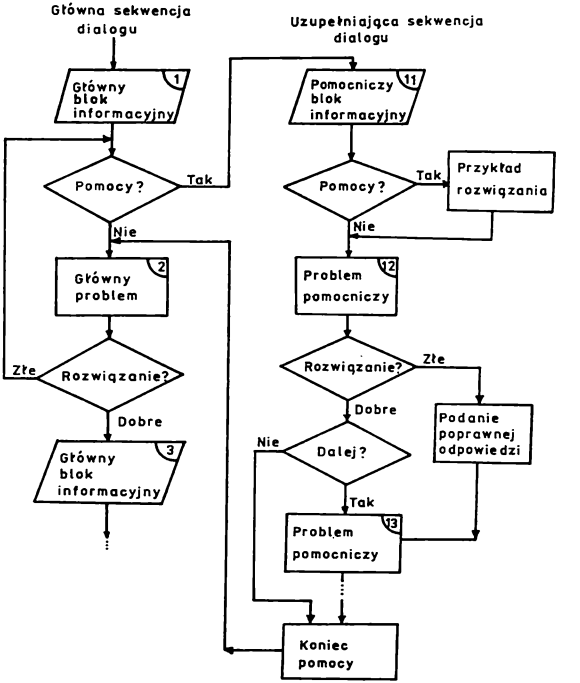
W komputerowych systemach uczących można wyróżnić kilka rodzajów dialogów. Jedną grupę stanowią dialogi występujące w głównej /podstawowej/ sekwencji materiału dydaktycznego. Do projektowania tej grupy dialogów służą dwie metody. Jedna polegająca na sekwencyjnym projektowaniu dialogu, "ekran po ekranie". Stosują ją przede wszystkim początkujący autorzy. Druga grupa nawiązuje do projektowania strukturalnego. Podejście strukturalne przejawia się zarówno w podziale hierarchicznym treści materiału, jak i podziale procesu nauczania, począwszy od elementów najbardziej ogólnych do najbardziej szczegółowych.

Celem drugiej grupy dialogów jest dostosowanie poziomu wiedzy studenta /bądź nadrobienie materiału/ do przyjętych wymagań w głównej sekwencji materiału. Dialogi te mogą stanowić niezależną część jednostki /mogą mieć postać np. wstępnych testów sprawdzających poziom studenta rozpoczynającego naukę/ lub mogą być związane z określoną treścią nauczania w postaci tzw. procedur pomocy. Na rys. 2 pokazano przykład powiązania głównej sekwencji materiału z pomocniczą. Zasadą jest to, że student otrzymuje tylko wtedy pomoc, gdy tego potrzebuje sam lub gdy system na podstawie błędnych odpowiedzi studenta decyduje o udzieleniu mu pomocy.

Następną grupą dialogów są dialogi testujące wiadomości studenta. W systemach nauczania wspomaganego komputerem testy nie tyle służą ocenianiu czy egzaminowaniu studentów, ile są pomocne głównie do wykrycia braków w wiadomościach studenta i zalecenia mu odpowiednich materiałów dydaktycznych, programów, skryptów itp.

Ostatnią z wyróżnionych grup dialogów są dialogi wspomagające studenta w dowodzeniu twierdzeń, przeprowadzaniu eksperymentów, wykonywaniu analiz itp. Ich zasadniczym celem jest pobudzenie studenta do kreatywnego myślenia i udzielania pomocy w samodzielnym rozwiązywaniu różnych problemów teoretycznych i praktycznych. W naukach ekonomicznych jedną z głównych możliwości zdobywania doświadczenia przez studenta jest eksperyment

[8] Niektóre jednostki dydaktyczne mocno związane z techniką przetwarzania informacji potwierdziły tę regułę.



Rys. 2. Ramowy schemat projektu dialogu w systemie nauczającym

przeprowadzony na modelu, który w przypadku dużej złożoności może być rozwiązany tylko na maszynie. W tych zadaniach student powinien uzyskać pomoc głównie w modelowaniu obiektów ekonomicznych i w analizie funkcjonowania modelu.

4. Rozplanowanie informacji na ekranie

Ogólne zasady prezentacji tekstu w komputerowych systemach uczących są podobne do reguł stosowanych podczas redagowania maszynopisu. Wyświetlany materiał powinien być atrakcyjny i łatwy do czytania, szczególnie zaleca się schematy, obrazy, grafiki, które skracają czas opanowania materiału. Tekst powinien być tak rozmieszczany, aby czytanie przebiegało od lewej do prawej strony i od góry do dołu. Powinno się unikać długich wielowierszowych paragrafów, należy dzielić je na mniejsze części i przedzielać jednym lub kilkoma wolnymi wierszami. Marginesy z prawej i lewej strony tekstu powinny być takie same w całej jednostce. Należy unikać pisania od brzegu do brzegu ekranu, ponieważ skrajne pozycje nie są dobrze widoczne. Ponadto tekst o długich wierszach /78 znakowych/ jest męczący dla oczu i trudny do czytania. Należy również pamiętać, że zbyt szerokie marginesy "rozbijają" treść zdań i utrudniają rozumienie materiału. Dotyczy to też górnego i dolnego wiersza ekranu. W projektowaniu tekstów materiału dydaktycznego w systemie OSKAR przyjęliśmy wiersze 40-60 znakowe, a marginesy miały od 5 do 15 znaków. Na ekranie mającym 24 wiersze zaleca się umieszczanie tekstu w wierszach od drugiego do dwudziestego pierwszego, gdyż system operacyjny wykorzystuje 2 ostatnie wiersze /23, 24/ na komunikaty i informacje.

Nie należy projektować przepełnionych ekranów, lepiej jest uzielić materiał na kilka kolejnych obrazów; przy czym stare teksty należy wycierać i na ich miejsce wprowadzać nowe informacje. Nie zaleca się dopisywania ich do istniejących fragmentów tekstu. Słownictwo i struktura zdań powinna być w miarę możliwości jak najprostsza. Jeśli student musi czytać tekst dwukrotnie, to jest to sygnał do przeredagowania tekstu. W projektowaniu dialogu przyjęto zasadę, że student steruje czasem wyświetlania tekstu oraz szybkością jego czytania.

Istotniejsze fragmenty prezentowanego tekstu powinny być wyróżnione, np. przez podkreślenie, obramowanie, większe wcięcie, pisanie dużymi literami lub rozstrzelenie liter. Niemożliwe jest natomiast stosowanie różnych kolorów, rozmiarów i krojów wyświetlanych znaków[9].

Teraz skoncentrujemy się na projektowaniu procesu interakcji studenta z systemem i związanych z tym metod stawiania pytań i analizy odpowiedzi.

5. Projektowanie pytań i antycypowanie odpowiedzi

Jak pokazano na rys. 2, w dialogu student-system należy zaprojektować główną treść informacyjną oraz rozwiązać dwa problemy. Pierwszy jest związany z projektowaniem procedur pomocy systemu wywoływanych wtedy, gdy student ma trudności bądź ze sterowaniem procesem nauczania, bądź też z obsługą systemu. Autor musi zatem zaprojektować dialogi z informacją uzupełniającą, pomagającą w zrozumieniu tekstu głównego, przykładami i ćwiczeniami o mniejszym stopniu trudności, szczegółowymi instrukcjami postępowania bądź wyjścia z "martwego" punktu. Autorzy powinni raczej unikać "karania" studenta za korzystanie z pomocy. Drugi problem odnosi się do zagadnień sterowania procesem dydaktycznym, zwłaszcza konstrukcji pytań i sprzężeń zwrotnych. Autor nie powinien stwarzać sytuacji, w których student musi odgadnąć autorską postać prawidłowej odpowiedzi. Jest to niepożądane, gdy odpowiedzi prawidłowych jest wiele. Należy też unikać pytań, które wymagają więcej niż jednej odpowiedzi. W celu ilustracji problemu zanalizujmy przykładowe pytanie:

Z banku pobrano do kasy 12000 zł na wypłatę wynagrodzeń.

1/ Zapisz tę operację gospodarczą...

Jeśli stawiamy w ten sposób pytanie, to musimy rozbudować procedurę analizy odpowiedzi. Zauważmy, że możemy spodziewać się wielu postaci prawidłowych odpowiedzi[10]. Ponadto odpowiedź stu-

[9] Większość systemów nauczania wspomaganego komputerem korzysta z monitorów graficznych, pozwalających na wprowadzenie szerokiej gamy obrazów graficznych, koloru i ruchu. Patrz [1].

[10] Przykłady prawidłowych odpowiedzi: 1/Kasa/Rach.rozlicz. 12000,-, 2/Kasa Wn,Rach.rozliczeniowy Ma, 12000,- 3/Kasa Dt 12000, Rach.rozliczeniowy Ct,12000,- 4/Kasa obciążyć,Rach.rozliczeniowy uznać 12000,- 5/Kasa w dobro konta, Rach.rozliczeniowy w ciężar konta.

denta może przybrać postać jednej odpowiedzi złożonej lub sekwencji odpowiedzi częściowych[11]. Widzimy zatem, że ten sposób stawiania pytań jest skomplikowany, zwiększa program uczący, wydłuża czas analizy, a także wprowadza studenta w zakłopotanie, gdyż zastanawia się nad odgadnięciem odpowiedzi. Ten sam efekt dydaktyczny osiągniemy, gdy to pytanie rozłożymy na dwa pytania proste:

1/ Podaj nazwę konta, które należy obciążyć. ... , i po prawidłowej odpowiedzi:

2/ Bardzo dobrze! A teraz podaj nazwę konta przeciwstawnego. ...

W pytaniach z ograniczonym wyborem, w których odpowiedź ma być podana jako: a, b, c, d lub e należy przewidzieć sytuację, że student odpowie np. a. lub a/ bądź a lub 1. Nie można w tym przypadku traktować tylko jednej z tych postaci odpowiedzi jako poprawnej. Autor musi antycypować także pewne powszechne błędy lub notacje i przewidzieć dla nich specjalne odpowiedzi systemu, np. w sytuacji zapisu tej samej kwoty: 2150.00, 2.150 lub 2150,-

Autor powinien zróżnicować reakcje systemu na kolejne błędy studenta; nie może ujednolicać odpowiedzi czy powtarzać w pętli tego samego komentarza systemu, czy wreszcie eliminować studenta z systemu. Przyczyną powtarzającego się błędu może być przecież źle sformułowane pytanie. W pytaniach o odpowiedziach dwuwartościowych lub z wielowyborem nie należy dawać szansy na powtórną odpowiedź, ponieważ w ten sposób można "znaleźć" poprawną odpowiedź. Lepiej w takich przypadkach skomentować złą odpowiedź lub podać sposób dojścia do poprawnej odpowiedzi.

W analizie odpowiedzi tekstowych należy też przewidzieć możliwość wystąpienia błędu ortograficznego, błędnego naciśnięcia klawiszy itp. W tych sytuacjach należy studentowi dać szansę poprawnego zapisu odpowiedzi. Gdy oczekiwane odpowiedzi przyjmują wartość numeryczną, autor powinien przewidzieć różne postacie zapisu liczb, określić obszar wartości i tolerancję

[11] W razie niepełnej odpowiedzi trzeba przewidzieć sekwencję instrukcji czytania przedzielonych analizą odpowiedzi.

dopuszczalnego błędu, a w przypadkach skomplikowanych obliczeń skłonić studenta do korzystania z trybu OBLICZENIA lub PROGRAMOWANIE.

6. Zalecenia

Na zakończenie podamy kilka praktycznych wskazówek przygotowania materiału dydaktycznego w systemach nauczania wspomaganego komputerem.

Zasadnicza rada to: staraj się w pełni wykorzystać własności systemu komputerowego - dużą pamięć, szybkość działania, cierpliwość itp. Pamiętaj, że sposób prezentowania materiału powinien być dostosowany do możliwości komputera, zarówno w sensie sprzętowym, jak i programowym. Przy wyborze tematyki materiału dydaktycznego można się spotkać z dwoma podejściami. W pierwszej metodzie mówi się: wybierz temat związany z twoimi badaniami i nie zastanawiaj się do czasu ich ukończenia, gdzie jednostka będzie wykorzystywana[12]. W drugiej metodzie autorzy mają zakreśloną tematykę i strukturę materiału dydaktycznego. Na podstawie kilkuletnich doświadczeń możemy sądzić, że pierwszą metodę mogą stosować autorzy mający duże doświadczenie w korzystaniu z techniki komputerowej w nauczaniu. Z drugiej metody powinni korzystać autorzy początkujący bez doświadczenia informatycznego. Następnie wskazówki dotyczą organizacji procesów projektowania, testowania i oceny.

Projekt dialogu rozwiązuj według metodyki projektowania strukturalnego.

Przed ukończeniem jednostki nie dopuszczaj - jeśli to nie jest konieczne - do zmiany koncepcji, przeszkadzania sobie itp. Pamiętaj, że każdy nauczyciel ma własną metodę nauczania, którą uważa często bezkrytycznie za najlepszą.

Na podstawie doświadczeń stwierdziliśmy, że poddawanie jednostki testowaniu przez studentów przed zakończeniem projektu jest stratą czasu.

Zdecydowanie broń swojego programu. Zdarza się bowiem, że duża część uwag i krytyk powoduje jedynie pracochłonne, a mało

[12] Według tej reguły postępowania zaprojektowano większość jednostek dydaktycznych w systemie PLATO. Obecnie system ten zawiera kilkanaście tysięcy godzin materiału dydaktycznego, z około 200 dyscyplin. Patrz [1].

istotne zmiany. Niektóre z nich wynikają jedynie z chęci zademonstrowania odmiennego stanowiska opiniodawcy.

W trakcie testowania jednostki wielu początkujących autorów jest zdziwionych niezaradnością bądź niewiedzą studentów, którzy nie są w stanie rozwiązać elementarnych zadań lub zrozumieć oczywistych dla autora informacji /często nie podanych explicite/. Taką sytuację trzeba traktować jako błąd autora. Nie wolno zaś winić studentów.

Przygotuj się, że niektórzy studenci będą się skarżyć na rozwlekłość i nieatrakcyjność materiału, który poznali wcześniej lub opanowali szybciej w inny sposób. Takie stwierdzenia należy uznać za normalne, jeśli liczba krytycznych uwag tego typu jest nieznaczna.

Wymienione uwagi i postulaty nie mogą uchronić systemu przed bardzo prostymi i nieefektywnymi /względem wykorzystywanego środka/ programami uczącymi. Trzeba pamiętać o tym, że zachwyt i inicjatywa nauczycieli nie zawsze wystarcza do usprawniania procesu dydaktycznego. Dlatego nie należy oczekiwać, że słaby nauczyciel wspomagany bardzo dobrym komputerem z bogatym oprogramowaniem opracuje znakomity materiał dydaktyczny, zaprojektuje dobry dialog studenta z systemem.

LITERATURA

[1] D.L. B i t z e r, The Wide World of Computer-based Education, Advances in Computers, 15, Academic Press, 1976, s. 239-283.

[2] A. B o r k, Preparing Student-Computer Dialogs - Advice to Teachers, Irvine 1979, Educ. Techn. Center. Univ. of California.

[3] A. B o r k, Textual Taxonomy, Irvine 1981, Univ. of California,/maszynopis/.

[4] A. B o r k, B.L. K u r t z, An ADT Model for the Production of Computer Based Learning Materials, Irvine 1980, Educ. Techn. Center, Univ. of California.

[5] J. K o r c z a k, B. M a k o w s k a, Weryfikacja materiału dydaktycznego w komputerowych systemach uczących /w niniejszym zeszycie/.

[6] J. K o r c z a k, U. M r ó z, Podejście do projektowania jednostek dydaktycznych w systemie wspomaganego nauczania OSKAR, INFOGRYF'80, Kołobrzeg 1980, s. 119-128.

[7] J. K o r c z a k, P. P a p s t, Mensch-Maschine Dialog um CUU-System OSKAR, Mat.Konf.TU, Drezno 1982.

[8] J. K o r c z a k, J. W o j d y ł a, System wspomaganego komputerem nauczania OSKAR, [w:] Informatyka, Rachunkowość, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 181, Wrocław 1981, s. 115-125.

[9] J. M a r t i n, Dialog człowieka z maszyną cyfrową, Warszawa 1976, WNT.

[10] H.F. O'Neil i in., Issues in Instructional Systems Development, New York 1979, Academic Press.

[11] Technical/Mechanical Standards for Supported PLATO Lessons, CDC, Pub.No 76360818C, 1981.

CONVERSATION DESIGN IN COMPUTER-AIDED TEACHING SYSTEMS /AS BASED ON THE EXAMPLE OF THE OSKAR SYSTEM/

Summary

The article presents the principles of conversation design in computer-aided teaching systems, based on the experience of preparing didactic materials in the OSKAR system. The given proposals are illustrated with diagrams of the teaching process organization. Several examples of dialogues and some practical directions for the authors of teaching programs are also comprised in the paper.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Jerzy K o r c z a k
Barbara Ł u k a s i k-M a k o w s k a
Akademia Ekonomiczna we Wrocławiu

WERYFIKACJA MATERIAŁU DYDAKTYCZNEGO W KOMPUTEROWYCH SYSTEMACH UCZĄCYCH

W artykule zaprezentowano propozycje zakresu, technik i metod prowadzenia weryfikacji materiałów dydaktycznych dla unikalnej w kraju klasy informatycznych systemów uczących. Przedstawiono doświadczenia zebrane w toku projektowania i wdrażania jednostek dydaktycznych dla systemu nauczania wspomaganego komputerem OSKAR[1].

Głównym celem weryfikacji materiału dydaktycznego jest poprawa jego walorów merytorycznych i dydaktycznych, co pozwala na podniesienie jakości oraz efektywności nauczania. W toku tych prac autorzy jednostek dydaktycznych doskonalą kwalifikacje oraz zostają uściślone teoretyczne zasady projektowania materiału dydaktycznego.

W jednostkach dydaktycznych systemu OSKAR weryfikuje się:

- zakres merytoryczny jednostki /tematykę, strukturę i szczegółowość/,
- projekt dialogu studenta z systemem,
- zastosowaną technikę programowania,
- dokumentację projektowo-programową jednostki.

Te cztery aspekty weryfikacji są ściśle z sobą powiązane i często granica między jednym obszarem a pozostałymi jest niewyraźna. To przenikanie powoduje, że zmiany w jednym z obszarów powodują zazwyczaj modyfikacje pozostałych, z tym że konsekwencje takiego działania nie zawsze są korzystne dla całości.

Stawia to szczególnie trudne zadanie przed zespołem osób weryfikujących. W systemie OSKAR zakładano, że zespół będzie

[1] Realizację systemu OSKAR zapoczątkowano w ramach problemu resortowego RI-14 w 1979 roku w Akademii Ekonomicznej we Wrocławiu.

zróżnicowany, a w jego skład wejdą: specjaliści z dyscypliny nauki, której dotyczy materiał dydaktyczny, pedagodzy z dobrą znajomością problemów dydaktyki, informatycy umiejący projektować systemy konwersacyjne i studenci, dla których jest przeznaczony materiał dydaktyczny. Ścisłe wytyczenie zakresu weryfikacji prowadzonej przez poszczególnych członków zespołu jest jednak bardzo trudne.

Weryfikacja jednostki dydaktycznej polega na jej zrealizowaniu przez określoną osobę z zespołu weryfikującego. Zakłada się, że oceny poszczególnych członków zespołu będą niezależne i dokonane bez udziału autora jednostki. Należy jednocześnie zwrócić uwagę, że jednorazowa realizacja jednostki nie zapewnia pełnej oceny. Konieczne jest bowiem przejście wszystkich przewidzianych przez autora rozgałęzień, sprawdzenie działania pętli w odniesieniu zarówno do poprawnych, jak i błędnych odpowiedzi, a także podjęcie próby "oszukania" systemu przez działanie niezgodne z zalecanym przez autora. To wszystko powoduje, że konieczne jest co najmniej trzykrotne przejście jednostki. Pierwsze w celu ogólnego zorientowania się w jednostce i zanotowania głównych, zwracających uwagę niedostatków /lub, przeciwnie, elementów szczególnie atrakcyjnych/. Drugie, bardziej szczegółowe, pozwalające zwrócić uwagę na rozwiązania przyjęte w projekcie, a wreszcie trzecie, w którym można przyjrzeć się fragmentom jednostki budzącym wątpliwości.

W praktyce działanie poszczególnych członków zespołu weryfikującego będzie jednak różne. Specjalista z danej dziedziny nauki ogranicza się zazwyczaj do sprawdzenia poprawności merytorycznej prezentowanych w jednostce zagadnień oraz zwrócenia uwagi na rozgałęzienia programu w przypadku poprawnych i niepoprawnych odpowiedzi. Studenta natomiast interesuje wyłącznie problem użyteczności jednostki, tzn. czy można z jej pomocą nauczyć się określonego zagadnienia i czy czas poświęcony na tę naukę będzie efektywniej wykorzystany niż w przypadku tradycyjnych metod uczenia się. Najbardziej wnikliwej i wielostronnej oceny będą natomiast dokonywać pedagodzy i informatycy. Spoczywa na nich odpowiedzialność nie tylko za formalną i merytoryczną poprawność materiału dydaktycznego, lecz również za komunikatywność dialogu oraz zaspokajanie potrzeb, a nawet gustów użytkowników materiału dydaktycznego - studentów.

Osiągnięcie takiego efektu jest możliwe przez szczegółowe poznanie wszystkich możliwości zaprojektowanej przez autora jednostki dydaktycznej, a następnie wskazanie błędów i nieprawidłowości oraz zaproponowanie modyfikacji określonych jej fragmentów. Wśród wymienionych problemów dwa pierwsze stanowią niejako zewnętrzny przejaw działania programu i od nich zatem należy rozpocząć weryfikację, przeprowadzając dla każdej jednostki dydaktycznej:

- analizę i ocenę założeń i koncepcji jednostki,
- ocenę przyjętych metod nauczania oraz
- określenie jakości zredagowania jednostki dydaktycznej.

W ramach pierwszej grupy należy ustalić zamierzenia, cele i intencje autora określające przeznaczenie jednostki dydaktycznej, następnie sprawdzić, czy są zgodne z programem dydaktycznym. Wprawdzie założenia te są zazwyczaj ustalane z autorem, gdy powstaje koncepcja jednostki, ale jednocześnie efekt końcowy często nie pokrywa się z zamierzeniami, co wynika najczęściej z braku doświadczenia autorów w opracowywaniu konwersacyjnych materiałów uczących. Następnie należy sprawdzić, jaki minimalny poziom wiedzy studentów założył autor opracowując materiał. Istotne jest ustalenie, czy autor konsekwentnie przestrzega przyjętych założeń, bo jeśli np. jednostka ma charakter wstępny, to trudno w trakcie jej realizacji wymagać od studentów wiadomości specjalistycznych lub szczegółowych. Jeśli zaś materiał dydaktyczny adresowany jest do studentów posiadających wiadomości podstawowe, wówczas wyjaśnienia uzupełniające muszą być bardziej "fachowe". Zaznaczmy, że rozbudowywanie jednostki "w głąb" powoduje rozszerzanie programu i najczęściej wiąże się z nieefektywnym zajęciem pamięci. Jako rozwiązanie prawidłowe należy wówczas zalecić rozdzielenie wstępnej i specjalistycznej części materiału i umieszczenie ich w dwu różnych jednostkach, z których każda będzie informować studenta o istnieniu tej drugiej jednostki.

Problematyka jednostki musi odpowiadać zawartości i strukturze programu danego przedmiotu, należy jednak dbać o to, aby nie było to powiązanie na zasadzie odbicia lustrzanego. Zastosowanie komputera pozwala bowiem na prezentowanie złożonych problemów oraz na modyfikację tej prezentacji stosownie do potrzeb i możliwości percepcyjnych studenta. Tak więc należy oczekiwać, że sformułowany w jednostce dydaktycznej problem będzie pra-

widłowo rozwiązany dzięki odpowiedniemu oddziaływaniu na studenta. Efektem końcowym tej części weryfikacji może być propozycja rozszerzenia lub ograniczenia treści materiału. Zespół może sugerować powiązanie analizowanej jednostki z innymi funkcjonującymi w systemie lub wskazać na celowość opracowania innych form materiałów dydaktycznych, zapewniających lepszą realizację zamierzeń dydaktycznych.

Druga grupa problemów wiąże się bardziej z metodyką nauczania. Wprawdzie jej poziom w szkołach wyższych jest niższy niż w szkolnictwie podstawowym czy średnim, to jednak w odniesieniu do komputerowych systemów uczących można wskazać kilka zagadnień, na które należy zwrócić uwagę, decydują one bowiem o szybkości i efektywności nauczania. Należy zatem zastanowić się, czy rzeczywiście celowe jest wykorzystanie systemu komputerowego do tego, by nauczyć studenta określonego tematu. Jeśli jednostka dydaktyczna jedynie wyręcza skrypt i umożliwia przeglądanie regułek czy wyjaśnień w innej niż książkowa /sekwencyjna/ kolejności, sprawdza wiadomości za pomocą pewnych stałych /stereotypowych/ pytań lub pozwala studentowi na reagowanie według określonych schematów /bez potrzeby zadawania sobie trudu przemyślenia problemu/, wówczas taka strategia nauczania nie wykorzystuje faktycznie wszystkich złożonych i różnorodnych możliwości komputera. Takie jednostki dydaktyczne nużą studentów, zajmują się oni bowiem wyłącznie przeglądaniem kolejnych fragmentów tekstu, a odpowiedzi na pytania udzielają często korzystając z "pomocy" kolegów, którzy wcześniej poznali określoną jednostkę dydaktyczną. O sytuacji takiej można się przekonać obserwując grupę studentów realizujących określone tematy w ramach zajęć nieobowiązkowych. Każda jednostka powinna zainteresować i zachęcić do współpracy studentów, którzy mogą się różnić zarówno zdolnościami, jak i przygotowaniem. Trzeba zatem zwrócić uwagę, czy autor przewidział wystarczającą liczbę odpowiedzi prawidłowych, złych lub niekompletnych, by w ten sposób zróżnicować /zindywidualizować/ kolejne realizacje jednostki dla różnych studentów. Należy przeanalizować pytania i komentarze systemu nie tylko ze względu na ich stylistyczną i merytoryczną poprawność, ale i konsekwencję używanej terminologii i oznaczeń. Takie formułowanie pytań, które sugeruje inną niż zamierzona przez autora formę odpowiedzi, powoduje zniechęcenie studenta.

Odpowiada on bowiem zgodnie z postawionym pytaniem, a system ocenia jego odpowiedź jako złą. Ponadto trzeba przeanalizować zestaw ćwiczeń i przykładów, które mają ułatwić zrozumienie materiału. Należy preferować takie rozwiązania, w których student może rozwiązywać zadania na danych przez siebie wprowadzonych, przyczynia się to bowiem do jego zaktywizowania. Zadania stałe, realizowane przez system należy traktować wyłącznie jako formę prezentacji. W tej części oceny należy również przeanalizować konstrukcję i zasady przeprowadzania testów.

Zakłada się, że testy są pomocą głównie dla studenta, który chce sprawdzić, czy prawidłowo zrozumiał materiał. Test może służyć również systemowi do sterowania procesem dydaktycznym. Nie można zatem oceniać studenta na podstawie jednej nieprawidłowej odpowiedzi /może być to błąd powodowany np. naciśnięciem niewłaściwego klawisza/. Zaleca się, żeby na ocenę składało się kilka elementów zebranych w toku realizacji całej jednostki. Konstrukcja jednostki powinna umożliwić ocenę postawy studenta wobec systemu, tzn. rozróżnienie błędów przypadkowych od prób celowego działania niezgodnego z zaleceniami. Test może umożliwić wyeliminowanie studentów niedostatecznie przygotowanych do nauki lub takich, którzy nie czynią dostatecznych postępów w toku nauki. Powinno to być związane z zaleceniem poznania określonej literatury bądź przerobienia jednostek wstępnych danego tematu. Gdy jednostki są długie, wieloczęściowe, to student powinien mieć możliwość ponownego rozpoczęcia nauki tam, gdzie uprzednio przerwał pracę /np. z powodu awarii systemu lub braku czasu/. Powinna to zapewnić dostatecznie rozgałęziona struktura jednostki oraz statystyka pracy studenta prowadzona przez system, w której byłaby przechowywana pełna informacja o przebiegu nauki.

Ostatnim elementem tej grupy jest "niezawodność" jednostki dydaktycznej, która wyraża się w działaniu programu konwersacyjnego nawet w przypadku niezgodności typu, formatu lub zakresu wartości wprowadzanych przez studenta danych /odpowiedzi/. Oczywiście w takim przypadku autor powiniem w programie przewidzieć wykrycie kategorii błędu oraz odpowiedni komentarz. Ten element weryfikacji bywa często przez zespół weryfikujący pomijany, gdyż poszczególni członkowie zespołu uważają za niestosowne

działanie niezgodne z zaleceniami programu i zazwyczaj starają się przejść przez jednostkę robiąc jak najmniej błędów. Tymczasem w praktyce bardzo duża część błędów w odpowiedziach studentów wynika nie tyle z braku wiadomości czy niezrozumienia postawionego pytania, lecz z pewnej emocji towarzyszącej zwłaszcza pierwszym samodzielnym próbom uczenia się pod nadzorem komputera. Zjawisko to dotyczy najczęściej studentów, którzy nie mieli żadnych zajęć związanych z informatyką lub nie próbowali korzystać z systemu komputerowego. Popełniają oni nawet tak proste błędy jak mylenie znaków "O" /litera/ i "0" /zero/, stawianie "." jako znaku mnożenia zamiast "x" czy naciśnięcia "spacji" przed wprowadzaną odpowiedzią lub po niej. Jednostka dydaktyczna powinna mieć zatem mechanizmy pozwalające uniezależnić się w działaniu od tego rodzaju błędów.

Wreszcie ostatnia, trzecia grupa problemów dotycząca jakości zredagowania jednostki dydaktycznej podobna jest do zadań redaktora książki, skryptu czy artykułu. Jest to praca bardzo żmudna, gdyż należy zwrócić uwagę na gramatykę, stylistykę i ortografię wyświetlanych komunikatów, a także prześledzić ich wygląd graficzny, a w ramach tego marginesy, odstępy, akapity. Ekrany muszą być przejrzyste i łatwe w czytaniu, nie powinny być jednak nadmiernie rozproszone, bo to męczy wzrok i uniemożliwia skupienie uwagi. Interesujący wygląd ekranów uzyskuje się przez stosowanie różnej wielkości znaków, różnej ich jasności i kontrastu, a również przez rozstrzelenie tekstu, podkreślenie lub zaakcentowanie dodatkowymi znakami fragmentów szczególnie ważnych. Należy jednak przestrzec przed nadmiernym stosowaniem wymienionych środków, gdyż wówczas zamiast pełnić rolę ważnych akcentów zamazują główny, logiczny tok prezentacji materiału. Szczegółowej analizie musi być również poddana terminologia. Należy sprawdzić, czy autor stosuje ją konsekwentnie, czy instrukcje i oznaczenia w różnych fragmentach jednostki są zgodne itp. Szczególną uwagę należy zwrócić na skróty terminów. Student powinien wiedzieć o wprowadzeniu do tekstu pełnego brzmienia pojęcia i jego skrótu jako synonimu[2]. W jednostce nie można posługiwać się kilku skrótami lub kilkoma oznaczeniami dla tego samego terminu. Należy dbać, by we wszystkich jednostkach dydaktycz-

[2]Należy również odesłać go do istniejących w systemie zbiorów informacyjnych, w których terminologia musi być taka sama jak w jednostce.

nych wspólnego obszaru merytorycznego była jednolita terminologia. O jej prawidłowości powinien zadecydować specjalista danego problemu. Spośród propozycji autorów powinien wybrać najkorzystniejsze lub opowiedzieć się za terminologią najbardziej utrwaloną w literaturze fachowej.

Dalsze działania zespołu weryfikacyjnego skoncentrowane są na "wewnętrznych" elementach jednostki, tzn. na konstrukcji jej programu oraz opracowanej dokumentacji projektowo-programowej jednostki. W tych działaniach największe obowiązki spadają na specjalistów informatyków oraz osobę lub zespół pełniący rolę koordynatora lub administratora systemu.

Należy również zwrócić uwagę na rozmiary programu. Ze względu na obecne ograniczenia konfiguracji sprzętu zbyt długi program opóźnia czas reakcji dla wszystkich użytkowników systemu, a ponadto w przypadku awarii systemu powrót do przerwanej pracy wymaga długiego czasu. Poza ogólnie przyjętymi regułami programowania programy w zależności od języka, w którym zostały opracowane, powinny mieć postać możliwie zwartą, a komentarze czy oznaczenia powinny w przyszłości użytkownikom /nie będącym autorami programów/ umożliwiać zmiany, przeróbki itp.

Dokumentacja prac projektowo-programowych będzie się składać w odniesieniu do wszystkich jednostek dydaktycznych wyłącznie z czterech elementów, tzn.:

- karty jednostki dydaktycznej,
- konspektu jednostki dydaktycznej,
- schematu jednostki dydaktycznej,
- źródłowej wersji programu.

Zadaniem zespołu weryfikującego jest sprawdzenie istnienia tych elementów oraz ustalenie zgodności między źródłową wersją programu a jego wersją dostarczoną do realizacji. Na podstawie dotychczasowych doświadczeń należy stwierdzić, że forma dokumentowania materiału dydaktycznego nie zdała egzaminu. Wysokie koszty utrzymania aktualnej dokumentacji materiału dydaktycznego zmusiły autorów i administratorów systemu do zapisu w systemie, a jedynym dokumentem tradycyjnym będzie wydruk źródłowej wersji programu dydaktycznego.

Weryfikowanie jednostki powinno być zakończone recenzją, na którą złożą się uwagi wszystkich członków zespołu. Najlepiej przedstawić ją autorowi bezpośrednio podczas wspólnej sesji, na której można wskazać wszystkie nieprawidłowe działania jednostki.

W recenzji autor powinien uzyskać następujące informacje:

- wyszczególnienie dobrych rozwiązań w jednostce,
- wykaz błędów, niedociągnięć i problemów nie w pełni rozwiązanych lub udokumentowanych,
- propozycje alternatywnych rozwiązań,
- wykaz tematów, zadań i problemów, które powinny się znaleźć w jednostce,
- ocena jednostki w stosunku do innych jednostek w systemie.

Szczegółowe uwagi na temat postaci ekranów najlepiej zaprezentować autorowi w czasie ich wyświetlania, tak aby mógł je nanieść bezpośrednio na wydruk programu źródłowego. Istnieje również możliwość sporządzenia opinii pisemnej i załączenie w niej kopii ekranów z zaznaczonymi uwagami.

Reasumując należy stwierdzić, że weryfikacja jest pracą szalenie żmudną i czasochłonną, wymagającą od zespołu nie tylko fachowości, ale również spostrzegawczości i cierpliwości. Znamienne jest, że weryfikację formalną /stylistyczną, gramatyczną, pojęciową/ lepiej robią osoby nie znające tematyki jednostki. Na podstawie doświadczeń stwierdzono, że pracochłonność opiniowania jednostki pochłania dziesięciokrotnie lub nawet kilkunastokrotnie czas nauki w ramach tej jednostki.

Na koniec kilka drobnych uwag, które nasunęły się autorom w czasie weryfikacji różnych jednostek dydaktycznych. Można by je opatrzyć wspólnym pytaniem: jakie są przyczyny rozbieżności między dobrymi założeniami /projektowymi/ a słabą jednostką dydaktyczną?

Jako pierwszą można wymienić brak doświadczenia autora i to zarówno w konstruowaniu dialogowego materiału dydaktycznego, jak i w pisaniu programów konwersacyjnych. Opracowanie pierwszej /przez autora/ jednostki dydaktycznej pozwala mu zrozumieć specyfikę konstrukcji i przygotować takie dialogowe narzędzie nauki. Należałoby zalecić autorom, aby zapoznali się z kilku wskazanymi im jednostkami dydaktycznymi, stanowiącymi zarówno pozytywne, jak i negatywne wzorce. Pojawia się zatem problem: jak przygotować autorów do pracy nad jednostkami dydaktycznymi. Jest on zbyt obszerny, by go omówić w niniejszym artykule. Będzie on przedmiotem oddzielnego opracowania[3].

[3] Jednym z elementów tego przygotowania mogą być zasady projektowania dialogów dla systemów uczących przedstawione w artykule J. Korczaka "Projektowanie dialogu w komputerowych systemach uczących" zamieszczonym w niniejszym zeszycie Prac Naukowych.

Drugą przyczyną są nawyki związane z operowaniem takimi narzędziami projektowania, jak papier i ołówek /lub inne materiały piśmienne/. Autorzy jednostek rozplanowując ekrany robią to zazwyczaj za pomocą tych konwencjonalnych narzędzi. Powoduje to, że nie występują wówczas pewne, specyficzne dla ekranu efekty optyczne /silna jaskrawość znaków na ekranie w stosunku do tła oraz określona zawczasu wielkość znaków, która sprawia, że wypełnienie rysunku jest zupełnie odmienne niż na papierze/. Autor przy rozplanowywaniu posługuje się liternictwem uzależnionym od własnego charakteru pisma oraz formatu makiety /najczęściej innego niż rzeczywisty format ekranu/, a czasem również od tego, czym pisze /pióro, ołówek, flamaster/. Wszystkie te elementy oddziałują korzystnie lub negatywnie na wygląd ekranu. Ma to może nieco mniejsze znaczenie, gdy przedmiotem projektu jest wyłącznie tekst, ale większe wówczas, gdy są nim wzory, drobne elementy graficzne czy tabele. Tak więc w czasie weryfikacji warto czasem na sprawdzany ekran nanieść planowane poprawki, aby przekonać się, czy odniosą one zamierzony skutek. Na przykład zastąpienie krótszego słowa dłuższym może spowodować zaburzenie układu graficznego całego ekranu. Tekst z prawej strony przepełnionego wówczas wiersza należy przenieść do wiersza następnego. Jeśli ekran był już bardzo zapełniony, to wprowadzenie nowego tekstu może zmniejszyć odstępy między wierszami wprowadzone dla przejrzystości tekstu. Jeśli chciałoby się je zachować, wówczas część tekstu należy zapisać na nowej stronie. Autorom można polecić, aby próbnie rozplanowywali wprost na ekranie. Jest to korzystne, wszystkie bowiem modyfikacje, jak np. rozszerzanie, skracanie czy wymazywania wiersza, zmiany znaków w wierszu można realizować systemowo /korzystając np. z TSO i projektując ekran w trybie redakcji zbioru danych o nienumerowanych wierszach/. Zauważono, że wielu autorów z dłuższym stażem w tej dziedzinie próbnie rozplanowuje na ekranie bardziej złożone elementy graficzne w jednostce.

Problem trzeci to dopasowanie materiału dydaktycznego w jednostce do wiedzy i możliwości autora. Problem ten jest skomplikowany, trudny do przezwyciężenia. Wynika głównie stąd, że przygotowanie jednostki dydaktycznej pochłania wiele pracy i

i czasu, autor często skłonny jest uważać ją za tym trudniejszą, im bardziej się nad nią napracował. W praktyce jednak wkład wysiłku autora i stopień trudności jednostki są sprawami zupełnie od siebie niezależnymi. Autorzy muszą być przygotowani do tego, by cały ich wysiłek zmierzał do zainteresowania studenta problemem, wciągnięcia go do współpracy, umożliwienia mu eksperymentowania i symulowania pewnych zjawisk. Wszystko to wymaga ogromnego wkładu inwencji i energii w opracowanie procesu dydaktycznego innego nieomal dla każdego ze studentów, którzy będą się zgłaszać do systemu[4]. Jeśli zatem autor przewidywał w jednostce tylko dwie drogi przejścia /poprawną i złą/, to taki materiał zazwyczaj nie znajduje szerszego zainteresowania, nie stwarza bowiem możliwości indywidualizacji nauczania.

Problem czwarty wynika z podchodzenia do problemu dydaktyki zbyt serio. Autor uważa, że musi zrobić jednostkę "mądrą", merytorycznie poprawną, a zapomina o skłonnościach studentów do wesołości. Fachowcy twierdzą, że nauka przez zabawę jest skuteczniejsza, dlaczego więc nie wykorzystać tego faktu w odniesieniu do nauczania wspomaganego komputerem, tym bardziej że praca przy monitorze męczy, zwłaszcza gdy pozostajemy w skupieniu przez dłuższy czas. Warto zatem wprowadzić do jednostek pewne elementy gier, zróżnicować komentarze, aby były czasem złośliwe a czasem dowcipne, zadbać, aby elementy grafiki przyciągały wzrok.

Takich problemów dobrego projektowania materiału dydaktycznego można by wymienić jeszcze wiele, np. problem rzetelności studenta w czasie jego współpracy z jednostką. Problem ten wymaga jednak dalszych badań oraz zebrania doświadczeń ze współpracy ze studentami różnych lat i kierunków studiów, jak również z wieloletniej współpracy z określoną grupą studentów i to zarówno w dziedzinie informatyki, jak i innych dyscyplin nauki. Brakuje zwłaszcza doświadczeń w zakresie projektowania i wykorzystania materiałów dydaktycznych w odniesieniu do przedmiotów nieinformatycznych.

---

[4] Odbiciem tego problemu jest pracochłonność związana z przygotowaniem jednostki dydaktycznej. Początkujący autorzy na opracowanie i przestudiowanie muszą poświęcić od 100 do 200 godzin pracy, gdy tymczasem autorom doświadczonym zajmuje to zazwyczaj od 15 do 30 godzin.

Należy jednak pamiętać, że student uczestniczący w jednostce dydaktycznej konwersuje nie z komputerem, ale z konkretną osobą, autorem materiału. Od niego więc w głównej mierze zależy, czy dydaktyka wspomagana komputerem zostanie zaakceptowana przez studentów i przyczyni się do poprawy efektywności nauczania czy też, przeciwnie, zostanie odrzucona jak wiele dotychczas podejmowanych prób stosowania różnych modyfikacji i usprawnień nauczania.

VERIFICATION OF DIDACTIC MATERIAL FOR COMPUTER TEACHING SYSTEMS

Summary

Methods and techniques for verifiying of didactic material for informatic teaching systems are presented in the paper. The structure and scope of the verifying staff are discussed, as well as, the basic problems to be solved in preparing student material for active and effective learning.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Janusz W o j d y ł a
Akademia Ekonomiczna we Wrocławiu

INTERAKTYWNE LABORATORIUM WSPOMAGANEGO KOMPUTEREM PROJEKTOWANIA BAZ DANYCH

W minionej dekadzie lat siedemdziesiątych nastąpił postęp w tworzeniu i użytkowaniu baz danych. Osiągnięcia teoretyczne badaczy amerykańskich w zakresie modelowania struktur pamięci /E. Senko, D. Lefkovitz, V. Lum, A. Cardenas, B. Yao, D. Severance i inni/, projektowania logicznych struktur danych /P.P. Chen, J. Hoffer, T. Teorey, J. Fry, G. Hubbard, M. Mitoma i inni/ oraz definiowania potrzeb informacyjnych przyszłych użytkowników bazy danych /J. Bubenko, P. Chen, G. Everest, D. Teichroew, D. Ross i inni/ sprawiły, że poczyniono znaczny postęp w tworzeniu i doskonaleniu systemów zarządzania bazami danych, jak i użytkowaniu baz danych w informatycznych systemach zarządzania[1].

Byłoby błędem sądzić, że wspomniane osiągnięcia teoretyczne doprowadziły do odtworzenia struktury procesu projektowania baz danych począwszy od rozpoznania wymagań informacyjnych przyszłych użytkowników, a kończąc na opisie struktury pamięci, w której będą przechowywane dane. Obecna wiedza jest taka, że projektowanie baz danych jest procesem słabo ustrukturalizowanym, wykorzystującym wiele różnych, często naprędce wymyślonych, metod i narzędzi programowych.

Autor serdecznie dziękuje wszystkim koleżankom i kolegom z Instytutu Informatyki, Ośrodka Komputerowego oraz studentom za udział w tworzeniu i rozbudowie laboratorium. Szczególnie dziękuję Barbarze i Jackowi Kortom, Jerzemu J. Korczakowi, Stanisławowi Zimnosze, Wiesławowi Borowskiemu, Bronisławowi Wiśniewskiemu, Jerzemu Wojtunikowi oraz Adamowi Papstowi. Dziękuję również serdecznie pani doc. dr hab. Elżbiecie Niedzielskiej za opiekę i zainteresowanie pracami zespołu. Postać do-

[1] Przegląd osiągnięć w tworzeniu baz danych zawierają prace [7; 8; 22; 36].

kumentacji oraz jej czytelność jest zaś zasługą Haliny Będzińskiej i Elżbiety Kwaśnik.

W ostatnich latach nastąpił rozwój środków programowych, wspomagających te procesy. Wspomnijmy na przykład pakiety:

- słowników skorowidzów /np. UCC-10, DATAMANAGER, LEXICON/,
- oceny przepustowości systemu bazy danych /np. DBDE, ARTS/DB, FDA czy DBPROTOTYPE/,
- wspomaganego projektowania /np. DB-DSS, Database Designer/[2].

Były one jednakże rozwijane oddzielnie i posługiwanie się nimi w praktyce projektowej jest bardzo kłopotliwe. Za główne przyczyny trudności można uważać:

- brak integracji między różnymi pakietami programowymi,
- zmienność formatów parametrów,
- niezgodność wymaganych danych o projektowej bazie,
- brak możliwości zapamiętywania otrzymanych rozwiązań cząstkowych,
- złożony proces komunikowania się projektanta z programem wspomagającym.

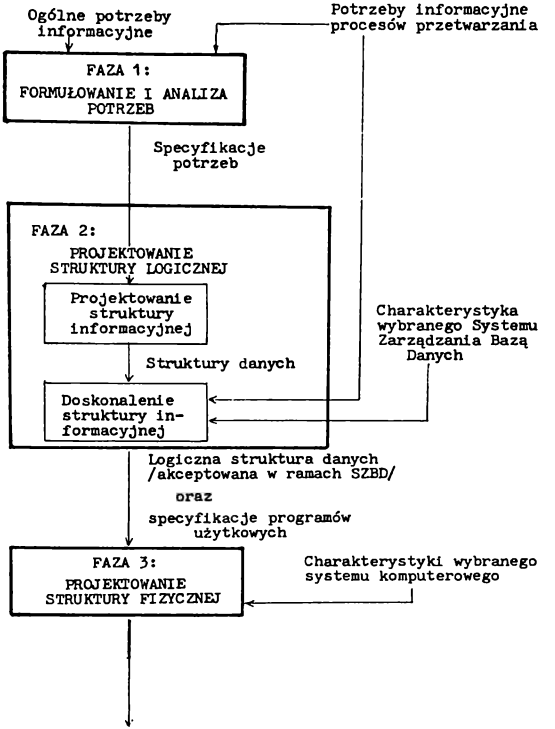
Po sprecyzowaniu istotnych wad aktualnie dostępnych zestawów programów wspomagających projektowanie baz danych przystąpimy do określenia wymagań, które musi spełniać pakiet programowy /a może nawet system informatyczny/ ułatwiający pracę analityka i projektanta bazy danych. Oto one:

1. Pakiet musi zapewniać wymienność danych wykorzystywanych przez różne programy w kolejnych fazach procesu projektowania bazy danych /patrz rys. 1/. Projektant musi być pewien, że jeśli wprowadzi określony parametr na potrzeby jednego programu, to wartość tego parametru będzie taka sama dla innych programów.

2. Należy przewidzieć możliwość wprowadzenia i przechowywania strumieni danych opisujących:

---

[2] Przegląd pakietów słownika /skorowidza danych zawiera książka H. Lefkovitza [16] oraz opracowania National Bureau of Standards [25] i [1]. Pakiety FDA oraz DBDE powstały w University of Michigan, Ann Arbor, [28] i [29]. Informacje na temat pakietu oceny organizacji baz danych typu CODASYL--ARTS/DB znajdują się w [11]. Pakiet DBPROTOTYPE oraz współpracujące z nim pakiety: IMS MAP, DCANALYZER, GPAR, IMS ASAP/II, IMSTRAP DL/I oraz DB/DC Data Dictionary opracowała firma IBM na potrzeby bazy danych IMS. Pakiet DB-DSS /Database Design Decision Support System/ opracował R. Gerritsen [9], a pakiet Database Designer: S. March [17].



Rys. 1. Fazy procesu projektowania baz danych /według J.P. Fraya, T.J. Teoreya i L.B. Oberlandera, [7, s. 22]/

- istniejące oraz przyszłe struktury informacyjne,
- procesy przetwarzania,
- wykorzystywany sprzęt wraz z oprogramowaniem systemowym i systemem zarządzania bazą danych.

Specyfikacje wymienionych strumieni powinny być w miarę możliwości gromadzone i wprowadzane automatycznie. Dotyczy to zarówno zbiorów danych /przez komputerową analizę ich struktur oraz wystąpień elementów/, jak i procesów /analiza programów źródłowych z punktu widzenia wykorzystywanych elementów struktur danych oraz złożoności technologicznej samego algorytmu bądź algorytmów przetwarzania/[3].

3. Programy wspomagające proces projektowania powinny umożliwiać analizę problemów zarówno w ujęciu ex ante, jak i ex post, gdyż doskonalenie zaprojektowanej i wdrożonej bazy danych należy uważać za nieodłączny element procesu projektowania sensu stricto. Tym samym zakres funkcjonalny systemu wspomagającego zostałby rozszerzony o fazę bieżącej eksploatacji i stanowiłby cenną pomoc dla administratora bazy danych. Gromadzenie w ramach jednego systemu danych projektowych oraz pomiarów eksploatacyjnych umożliwiałoby, w sprzężeniu zwrotnym, doskonalenie analitycznych lub symulacyjnych programów modelujących procesy zachodzące w rzeczywistej bazie danych.

4. Wywoływanie i wykonywanie programów wspomagających oraz dostęp do metabazy danych powinien być interaktywny, zorientowany na użytkownika systemu wspomaganego projektowania baz danych[4]. Można to osiągnąć dzięki dialogowi indagacyjnemu, transakcyjnej formie przetwarzania komunikatów oraz zdolności "podglądania" zbiorów metabazy.

5. Należy dążyć do tego, aby programy wspomagające były homogeniczne. Ponadto należy przewidzieć możliwość współpracy programów heterogenicznych oraz możliwość definiowania i przechowywania danych specyficznych dla jednego zastosowania lub użytkownika.

---

[3] Problemy komputerowej analizy zbiorów danych rozważano w pracach [5; 32].

[4] Pod pojęciem metabazy będziemy rozumieli bazę danych funkcjonującą w systemie wspomaganego projektowania o projektowanej bazie danych.

6. Programowa nadbudowa systemu interaktywnego projektowania baz danych powinna umożliwiać łatwą jego rozbudowę /dzięki wprowadzeniu nowych narzędzi wspomagających/ oraz eksploatację /dzięki funkcjonowaniu grupy programów organizacyjnych przeznaczonych między innymi do utrzymywania bibliotek programów, zbierania danych statystycznych, ochrony zbiorów metabazy, składowania, odnawiania/,

Wspomagane komputerem tworzenie systemów informatycznych /WKT SI/ może być realizowane za pomocą narzędzi programowych charakteryzujących się różnymi wartościami wskaźników integracji oraz elastyczności[5]. Integracja systemu WKT SI jest miarą /0-1/ zdolności realizacji wybranego etapu prac podstawowych /np. analizy, projektowania, programowania lub eksploatacji próbnej/. Podobnie integracja systemu interaktywnego projektowania baz danych będzie miarą zdolności realizacji wybranej fazy /patrz rys. 1/.

Elastyczność systemu wspomaganego komputerem tworzenia systemów informatycznych jest miarą /0-1/ zdolności realizacji struktury tworzonego systemu /np. informacyjnej, funkcjonalnej lub technologicznej/. Podobnie elastyczność systemu interaktywnego projektowania baz danych będzie miarą zdolności realizacji wybranej struktury tworzonego, w technologii bazy danych, systemu informatycznego /np. informacyjnej, funkcjonalnej /proceduralnej/ lub technologicznej/.

Każdą z wymienionych struktur można opisać i analizować w dwóch płaszczyznach /przekrojach/, tj. typów oraz wystąpień, jako całość lub jako część /element struktury/. Każda ze struktur składa się z pewnej liczby elementów powiązanych relacjami jakościowymi /nadrzędności, podrzędności i równorzędności/ oraz ilościowymi /wyrażającymi wzajemny stosunek elementów, tj. 1 do 1, 1 do N oraz M do N/. Każdy element będzie charakteryzowany za pomocą zespołu właściwych mu atrybutów. Nazwy elementów, relacji i atrybutów, charakterystyczne dla każdej z trzech wymienionych struktur bazy danych, zaprezentujemy oddzielnie.

Struktura informacyjna projektowanej bazy danych może być złożona z różnych elementów w różnych fazach jej powstawania,

[5]Opis miar integracji i elastyczności systemów wspomaganego komputerem tworzenia systemów informatycznych zawiera praca [34].

co przedstawia tabela 1. Należy wspomnieć, że niektóre elementy struktur, zawarte w tabeli 1, są już dzisiaj zbierane i przechowywane przez specjalistyczne pakiety programowe zwane słownikami /skorowidzami danych/[6]. Słownik/skorowidz jest sam w sobie systemem informacyjnym, którego baza danych zawiera definicje typów danych, procesów oraz związków między nimi, nie zawiera natomiast danych lub procesów dotyczących określonego zastosowania. Analizy porównawcze słowników/skorowidzów wykazują dużą różnorodność przechowywanych elementów struktur informacyjnych [por. 1;16;25]. Obecnie ze słowników/skorowidzów korzysta się jedynie w fazach projektowania struktur logicznych i fizycznych. Wyjątek stanowi słownik DDS ICL 2900, oferowany przez firmę brytyjską ICL, który umożliwia wspomaganie prac zmierzających do utworzenia modelu informacyjnego analizowanego obiektu [por. 31].

W celu wyraźnego zróżnicowania obiektu słownika od obiektu projektowanej bazy danych ten pierwszy będziemy określali mianem metaobiektu, np. metaobiekt dana, metaobiekt proces lub w skrócie m-dana bądź m-proces. Wszystkie metaobiekty będą charakteryzowane przez swoje atrybuty, np. atrybuty zewnętrzne / /pierwotne/, atrybuty wewnętrzne /wtórne, obliczane/ oraz atrybuty relacji między metaobiektami.

A t r y b u t y z e w n ę t r z n e muszą być wprowadzane do słownika z analizowanego lub projektowanego systemu konkretnego użytkownika, np. nazwa danej elementarnej, jej długość, precyzja, klasa dopuszczalnych wartości. A t r y b u t y wewnętrzne mogą być obliczone lub uzyskane na podstawie operacji wykonanych na zapamiętanych uprzednio atrybutach pierwotnych lub /oraz/ innych atrybutach wewnętrznych, np. częstość wykorzystania grupy danych w różnych procesach, rodzaj urządzenia, na którym jest przechowywany zbiór. A t r y b u t y r e l a c j i charakteryzują związki między metaobiektami jednego lub kilku typów. Atrybuty te są w swej istocie zawsze zewnętrzne lub wewnętrzne, ale rozpatrujemy je jako osobny typ ze względu na ich kluczową rolę w zapewnieniu integracji słow-

[6] Słownik/skorowidz /ang. Data Base Dictionary/Directory, w skrócie DBDD lub DDS - Dictionary/Directory System/ jest pakietem programowym, ułatwiającym opis i przechowywanie informacji o tworzonym systemie informatycznym. Znajduje zastosowanie w procesach tworzenia zarówno systemów tradycyjnych, jak i realizowanych w technologii bazy danych.

Tabela 1

Zawartość przykładowej metabazy systemu wspomagającego projektowanie baz danych

STRUKTURA	Fazy procesu projektowania bazy danych		
	formułowanie i analiza potrzeb	projektowanie struktury logicznej	projektowanie struktury fizycznej
informacyjna	Obiekty Relacje Atrybuty Wejście/wyjście	Schemat Podschemat Zbiór Rekord logiczny Dana grupowa Dana	Baza danych Obszar Strona Rekord fizyczny Pole grupowe Pole
funkcjonalna	Funkcje Procesy	Problem Operacja	Problem Zadanie Program Moduł
technologiczna		Użytkownik Punkt przeznaczenia Terminal logiczny Łącze Punkt kontaktowy	Wolumin Terminal fizyczny Urządzenie Operator

nika/ skorowidza, a w konsekwencji integracji projektowanej bazy danych.

Dotychczasowe prace nad rozwojem pakietów słowników/skorowidzów danych koncentrowano jedynie na typach elementów. Uniemożliwiało to, między innymi, ilościową analizę procesów odwzorowywania schematów konceptualnych w logiczne i fizyczne. Ponadto brak kwalifikacji danych utrudniał podejmowanie decyzji przez administratora bazy danych w takich aspektach jej technologii, jak:

- reorganizacja zbiorów i obszarów,
- wybór efektywnej metody kompresji,

- ocena przepustowości podsystemu telekomunikacji,
- oszacowanie podstawowych parametrów obszaru pamięci oraz czasu trwania wybranych procesów itp.

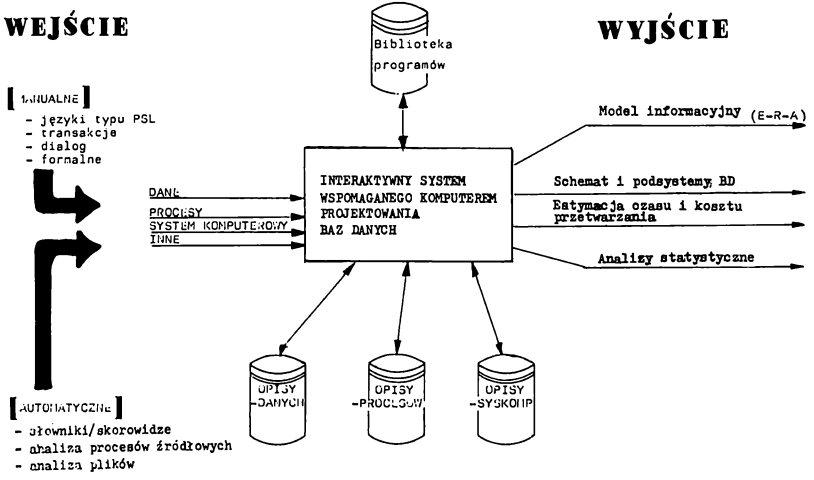
Po przyjęciu założenia, że strukturę informacyjną tworzonego systemu informatycznego należy analizować trójpoziomowo /co wynika z przyjętej metodyki, zaprezentowanej na rys. 1/ w Instytucie Informatyki Akademii Ekonomicznej we Wrocławiu opracowano koncepcję oraz częściowo wdrożono system wspomaganego komputerem projektowania baz danych [por. 21; 32].

Zgodnie z przyjętym założeniem tworzony system informatyczny powinien być analizowany na trzech poziomach, tj. konceptualnym /obiekty, relacje i atrybuty/, logicznym /obszary, rekordy, pola itp./ oraz fizycznym /woluminy, pliki, strony, bloki itp./. Tej trójpoziomowej analizie powinny jednocześnie towarzyszyć rozważania i ewentualnie obliczenia na dwóch płaszczyznach, z których pierwsza dotyczy typów, a druga wystąpień elementów wyróżnionych na wymienionych poziomach.

Opisy elementów należy wprowadzać w kolejności analizowanych struktur /tj. konceptualnej, logicznej i fizycznej/, a opis typów powinien zawsze poprzedzać opis wystąpień elementów jednego rodzaju. Jak przedstawiono na rys. 2, główne strumienie informacji wejściowych w interaktywnym laboratorium projektowania baz danych umożliwiają wprowadzanie zarówno opisów typów, jak i wystąpień elementów tworzonego systemu informatycznego. Wszystkie opisy elementów podzielono na grupy. Podstawą do klasyfikacji był udział w jednej z trzech podstawowych struktur systemu informatycznego, a mianowicie: informacyjnej /grupa-DANE/, funkcjonalnej /grupa-PROCESY/ oraz technologicznej /grupa-SYSTEM KOMPUTEROWY/. W celu umożliwienia dalszej, elastycznej rozbudowy wejścia utworzono czwartą grupę /INNE/ parametrów specyficznych dla konkretnego problemu lub przechowywanych przez krótki okres /np. w trakcie obliczeń/.

Wszystkie metadane wejściowe mogą być wprowadzane do metabazy ręcznie lub automatycznie. Jak dotychczas, automatyczne metody zbierania i wprowadzania metadanych są mało rozpowszechnione. Stosuje się je jedynie do opisów typów elementów[7]. Za-

[7] Przykładem mogą być programy analizy części deklaracyjnych danych w wersjach źródłowych programów lub schematów i podschematów baz danych, np. słownik/skorowidz DATAMANAGER/.



Rys. 2. Podstawowe strumienie informacyjne interaktywnego laboratorium projektowania baz danych

kładając, że wiele systemów informatycznych z bazą danych będzie wykonywanych w instytucjach lub przedsiębiorstwach, które już eksploatowały jakiś system, laboratorium wyposażono w pakiet automatycznej analizy plików utworzonych w standardzie systemu operacyjnego OS/JS. Pakiet o nazwie WAD umożliwia określanie parametrów struktury /na poziomie pliku, rekordu lub pola/, liczebności i dynamiki pliku dyskowego o dowolnej organizacji /sekwencyjnej i bezpośredniej, indeksowo-sekwencyjnej/[8]. Wspomniany pakiet programów jest stosowany również do symulowania kompresji danych oraz analiz statystycznych różnych wartości danych identyfikacyjnych i kwantyfikacyjnych[9]. Ponieważ prace nad automatyzacją wprowadzania opisów elementów metabazy dają interesujące wyniki, będą więc kontynuowane, zwłaszcza te, dotyczące uogólnionego analizatora sieciowych baz danych oraz analizatora procedur /w celu stworzenia narzędzia budowy modeli programów korzystających z opisanej bazy danych/[10].

Obecnie opisy tworzonego systemu wprowadza się najczęściej ręcznie do bazy. Aż do momentu upowszechnienia się fonicznych sposobów wprowadzania danych metodą manualną - realizowaną za pomocą formularzy, różnych języków opisu problemu /np. PSL, CSL, RPL/, dialogu konwersacyjnego bądź transakcyjnego - będzie się stosować w każdym metasystemie, w którym zajdzie potrzeba opisu nowo tworzonych elementów.

Problem wprowadzania opisów do metabazy należy do najważniejszych problemów tworzenia narzędzi wspomaganego komputerem projektowania nie tylko baz danych, ale również innych wytworów myśli ludzkiej. Dotychczasowe badania prowadzone w Instytucie Informatyki pozwoliły na sprecyzowanie kilku wniosków, dotyczących tego zagadnienia:

1. Z powodu niedoskonałości technicznej czytników dokumentów i wysokiego kosztu ich nabycia /a w warunkach krajowych

---

[8] Szczegóły konstrukcyjne zawiera opracowanie [21] oraz artykuł [32].

[9] Na bazie koncepcji automatycznej analizy plików przygotowano założenia konstrukcyjne kompresora danych, patrz [33, rozdz. 7].

[10] W założeniu powinien to być odpowiednik programu DBPROTOTYPE II opracowany przez firmę IBM na potrzeby baz danych IMS2 i IMS/VS.

także braku odpowiednich urządzeń/ wsadowe wprowadzenie danych z formularzy źródłowych opisu elementów metabazy jest pracochłonne i trwa zbyt długo. Zwykle od momentu wypełnienia formularza do czasu umieszczenia zawartych na nim opisów w metabazie mija zbyt wiele czasu /1-2 dni/. Wówczas projektant nie ma dostępu do najaktualniejszej dokumentacji.

2. Języki opisu problemu, a szczególnie język PSL, są zbyt złożonym narzędziem, zwłaszcza dla analityków systemu. Ponadto jego wsadowa realizacja ma wszystkie wady wymienione w punkcie 1. Wersje umożliwiające konwersacyjne wprowadzanie opisów wymagają wielu komputerów, co sprawia, że ich używanie jest kosztowne.

3. Wymienionych wad pozbawione są metody dialogowe o formie konwersacyjnej lub transakcyjnej, korzystające z integralnych monitorów telekomunikacyjnych systemów zarządzania bazami danych /np. IMS DB/DC lub IDMS-DC/ lub sprawdzonych i popularnych monitorów niezależnych /np. SHADOW, ROSCOE, CICS/.

4. Dalsza współpraca specjalistów z zakresu baz danych oraz sztucznej inteligencji doprowadzi w najbliższym czasie do rozwoju w zakresie wspomaganego projektowania - baz wiedzy, systemów wspomagania procesów podejmowania decyzji projektowych lub systemów typu EXPERT[11].

Powróćmy do koncepcji laboratorium, przedstawionej na rys.2. Kilka uwag poświęcimy organizacji metabazy. Obecnie wyodrębniono w niej trzy obszary grupujące rekordy z zakresu: struktur danych /OPISY-DANYCH/, procesów przetwarzania /OPISY-PROCESOW/ oraz systemu komputerowego wraz z charakterystyką systemu zarządzania bazą danych /OPISY-SYSKOMP/. Rekordy są następnie grupowane w zbiory danych o ustalonej strukturze i relacjach międzyrekordowych.

Prace nad konstrukcją zintegrowanego modelu danych metabazy laboratorium wspomaganego projektowania są znacznie zaawansowane. Coraz więcej doświadczeń dostarcza wykorzystywanie klasycznych struktur adresowanych bezpośrednio oraz struktur se-

[11] Prace tego typu prowadzi R. Gerritsen [9; 19]. Systemy wspomagające prace zawodowe typu EXPERT mają zdolność samosterowania oraz uczenia się podejmowania decyzji. Znalazły zastosowanie, między innymi, w medycynie - MYCIM [24], projektowaniu konstrukcji lotniczych - SACON [3] oraz diagnostyce chorób pasożytniczych roślin - PLANT [18;19].

kwencyjnych, udostępnianych dla programów napisanych w językach FORTRAN IV, PL/I oraz Assembler[12]. Implementacja opracowanego modelu nie będzie przedsięwzięciem prostym ze względu na systemy zarządzania bazą danych /tj. RODAN lub HADES ewentualnie nowo tworzony STEP-400/, z których można korzystać. Głównym problemem jest nieuwzględnienie języka FORTRANU IV, jako języka bazowego tych systemów, a właśnie w tym języku opracowano najwięcej programów analizujących lub symulujących procesy tworzenia i oceny baz danych. Należy zatem opracować makroinstrukcje języka DML dla FORTRANU IV lub przepisać programy w języku PL/I. To drugie rozwiązanie spowoduje zmniejszenie efektywności przetwarzania na komputerach o małej pamięci operacyjnej /256-512 K bajtów/. Dzięki użyciu języków DDL oraz DML nie trzeba będzie natomiast zbytnio dbać o zgodność formatów danych używanych przez programy napisane w różnych językach i w różnych ośrodkach projektujących. Kłopot może sprawiać nazewnictwo rekordów, plików, elementów danych itp. Można ich jednak uniknąć dzięki funkcjonowaniu w metabazie nazw zastępczych, tzw. aliasów[13].

Obecnie biblioteka programów obejmuje następujące pozycje:

1. FDA - analizator organizacji plików sekwencyjnych, indeksowo-sekwencyjnych, wielolistowych, indeksowanych /ze skorowidzem/ oraz bezpośrednich[14].

2. DIFI - analizator funkcjonowania plików przyrostowych[15].

3. METAOPIS - konwersacyjny program wprowadzania i aktualizacji opisów elementów metabazy[16].

---

[12] Konstrukcja modelu danych metabazy laboratorium wspomaganego komputerem projektowania danych stanowi zagadnienie, które autor artykułu porusza w przygotowywanej pracy habilitacyjnej.

[13] Nazwy tych samych elementów mogą mieć maksymalnie 6 znaków, w języku Assembler 8 znaków, a w języku PL/I 32 znaki.

[14] Analizator FDA /ang. File Design Analyzer/ otrzymano z Uniwersytetu w Michigan. Następnie dopisano moduły konwersacyjnego wejścia/wyjścia w języku polskim. Inne zastosowania oraz opis FDA, patrz [20; 27; 28].

[15] Pliki przyrostowe /ang. differential files/. Ich badaniem zajmował się D.Severance [23].

[16] Opis tych programów zawiera artykuł [32] oraz opracowanie [21].

4. WAD - zautomatyzowany analizator plików dyskowych[17].

5. BTREE - analizator funkcjonowania skorowidzów typu BtTree w bazach danych[18].

6. INTERS - Symulator interpolacyjnych metod przeszukiwania[19].

7. KOMPRAN - symulator algorytmów kompresji plików dyskowych.

8. PREZOPIS - program wyszukiwania i prezentacji opisów elementów metabazy.

9. PREZWAD - program prezentacji wyników pracy analizatora WAD w przekrojach: struktury, liczebności i dynamiki pliku lub wybranych jego elementów.

10. AIDEN - analizator identyfikatorów numerycznych[19].

11. ATEXT - analizator danych tekstowych[20].

12. ALOKATOR - wspomagany alokator obszarów dla plików sekwencyjnych, bezpośrednich oraz bibliotecznych w systemie OS/JS[21].

Obecne prace /marzec 1982/ zmierzają do zaprojektowania własnego pakietu słownika/skorowidza danych, uwzględniającego zaprezentowaną w artykule ideę trójpoziomowego odwzorowania danych na dwóch płaszczyznach, tj. typów oraz wystąpień elementów danych. Planuje się, że w roku następnym nastąpi wdrożenie pakietu z wykorzystaniem SZBD RODAN lub HADES z monitorem teleprzetwarzania SKOT. Kontynuację dotychczasowych badań z zakresu kompresji danych ma stanowić opracowanie projektu /do końca 1982/ oraz wdrożenie uniwersalnego pakietu kompresji danych KOMPRES [por. 36].

---

[17] Analizator BTREE opracował J. Korczak, patrz [14]. Obecnie opracowuje się rozszerzoną i udoskonaloną wersję.

[18] Symulator INTERS opracował S. Zimnocho, patrz [38].

[19] Analizator AIDEN opracowała E. Jankowska. Był to temat jej pracy dyplomowej /patrz [12]/. Autorka korzystała z badań przeprowadzonych przez J.Korczaka [13].

[20] Analizator ATEXT opracowała E.Krotosik w ramach pracy dyplomowej /patrz [15]/. Wykorzystała ona badania metod kompresji danych za pomocą n-gramów [33].

[21] Wspomagany komputerem alokator plików dyskowych opracowały B. Iwasieczko i B. Mazur, studentki Wydziału Zarządzania i Informatyki, w ramach prac seminarium "Wspomagane komputerem tworzenie systemów informatycznych", które prowadził autor w roku akademickim 1981/1982.

Na podstawie pracy doktorskiej B. Yao oraz obecnej wersji programu FDA planuje się skonstruowanie analizatora metody dostępu VSAM oraz symulatora reorganizacji plików dyskowych na podstawie algorytmów dynamicznej metody reorganizacji plików [por. 27;35;37].

Przewiduje się, że nadal będą trwały prace nad doskonaleniem programów BTREE oraz INTERS.

LITERATURA

[1] A Survey of Eleven Government-Developed Data Element Dictionary/Directory Systems. NBS 500-16, Washington 1977.

[2] H. A g h i l i, D.G. S e v e r a n c e, A. Practical Guide to the Design of Differential Files for Recovery on-line Databases. The University of Michigan, Ann Arbor, November 1980.

[3] J. B e n n e t, L. C r e a r y, R. E n g l e m o r e, R. M e l o s h, SACON: A Knowledge-based Consultant for Structural Analysis. Computer Science Dep., Stanford University, STAN-CS-78-699.

[4] J. B o t w i c k, R. H a a n, The Database Design Evaluator. Part II: The User Interface. Data Translation Project, The University of Michigan, Ann Arbor 1977.

[5] A.F. C a r d e n a s, Evaluation and Selection of File Organization. A Model and System,[w:] Comm. of ACM, 1973, nr 9, s. 540-548.

[6] Data Catalogue 2. A System Overview. Synergetics Corporation, One DeAngelo Drive, Bedford. Mass 01730. Bedford 1978.

[7] J.P. F r y, T.J. T e o r e y, L.B. O b e r l a n d e r, Database Design Tools and Methods. A Survey and Tutorial. The University of Michigan, Ann Arbor. Technical Report 78 DB 14.

[8] J.P. F r y, T.J. T e o r e y, Design and Performance Tools for Improving Database Usability and Responsiveness,[w:] Databases: Improving Usability and Responsiveness. Ed. by B. Schneiderman, Academic Press, New York 1978.

[9] T. G a m b i n o, R. G e r r i t s e n, A Database Design Decision Support System. Working Paper 77-04-12, Dept. of Decision Sciences, Wharton School, University of Pensylwania 1977.

[10] T. G a m b i n o, R. G e r r i t s e n, A Data Base Design Decision Support System,[w:] Proc. Very Large Data Bases, 1977, Tokyo, s. 534-544.

[11] C. H u l t e n, L. S o d e r l u n d, A Simulation Model for Performance Analysis of Large Shared Data Bases. Proc. of the Third Inter. Conf. of Very Large Data Bases, ACM, New York 1977, s. 524-532.

[12] E. J a n k o w s k a, Konstrukcja i zastosowanie analizatora identyfikatorów numerycznych w bazach danych. Praca dyplomowa napisana po kierunkiem J. Wojdyły. Akademia Ekonomiczna we Wrocławiu, Wydział Zarządzania i Informatyki, 1982.

13 J.J. K o r c z a k, Metoda analizy identyfikatorów zapisów w zbiorach o organizacji zrandomizowanej, w: Systemy wielodostępne, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 102, Wrocław 1977, s. 137-151.

14 J.J. K o r c z a k, Projektowanie struktury fizycznej indeksu B-TREE, [w:] Przegląd projektowania systemów, red. W.Olejniczak. Materiały konferencji INFOGRYF 80. Politechnika Szczecińska oraz TNOiK, Szczecin - Kołobrzeg 1980.

15 E. K r o t o s i k, Wspomagana komputerem analiza danych tekstowych z punktu widzenia możliwości ich kompresji. Praca dyplomowa napisana pod kierunkiem J.Wojdyły. Akademia Ekonomiczna we Wrocławiu, Wydział Zarządzania i Informatyki, 1982.

[16] H. L e f k o v i t z, Data Dictionary Systems. QED Information Sciences, Inc., Wellesley, Massachusetts, 1977.

[17] S.T.M a r c h, D.G. S e v e r a n c e, A Mathematical Modeling Approach to the Automatic Selection of Database Designs. Prac. of the ACM-SIGMOD Inter. Conference 1978, s. 52-65.

[18] R.S. M i c h a l s k i, R.L. C h i l a u s k y, Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis, [w:] "International Journal of Policy Analisis and Information Systems", 1980, nr 2, s. 125-161.

[19] R. M i c h a l s k i, Knowledge Acquisition Trough Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data Into Conjunctive Concepts. The University of Illinois, Departament of Computer Science UILO-ENG-80-1718.

[20] G. P o s a d a, User Manual for the FDA. University of Michigan 1980.

[21] Projekt komputerowego wspomagania systemu operowania i utrzymywania baz danych. Praca badawcza wykonana pod kierunkiem J.Wojdyły. Akademia Ekonomiczna we Wrocławiu, 1981.

[22] Projektowanie systemów informatycznych wspomagane komputerem, cz. I - Analiza możliwości. Praca badawcza wykonana pod kierunkiem J.Wojdyły. Akademia Ekonomiczna we Wrocławiu, 1980, /maszynopis/.

[23] D.G. S e v e r a n c e, G.M. L o h m a n, Differential Files: Their Application to the Maintenance of Large Databases. "ACM Trans on Database Systems", 1976, nr 3, s. 256-267.

[24] E.H. S h o r t l i f f e, R. D a v i e s, Some Considerations for the Implementation of Knowledge-Based Expert Systems. The MYCIN Project. SIGART Newsletter, December 1975, s. 9-12.

[25] Technical Profile of Seven Data Element Dictionary/Directory Systems. NBS 500-3, Washington 1975.

[26] D. T e i c h r o e w, E.A. H e r s h e y III, PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems. [w:] IEEE Trans. on Software Eng. 1977, nr 1, s. 41-48.

[27] T.J. T e o r e y, Detailed Specifications for the File Design Analyzer. Systems Engineering Laboratory Tech. Report 87, Department of Electrical and Computer Engineering, University of Michigan 1975.

[28] T.J. T e o r e y, K.S. D a s, Application of An Analytical Model To Evaluate Storage Structures. Proc. a of the Inter. ACM-SIGMOD Conf. 1976, s. 9-20.

[29] T.J. T e o r e y i inni, The Database Design Evaluator. Working Paper 77 DA 7.2. Data Translation Project, The University of Michigan, Ann Arbor 1977.

[30] UCC TEN. Data Dictionary/Manager. Concepts and Facilities. University Computing Company UCC Tower/Exchange Park, Dallas, Texas 75235 Dallas 1980.

[31] A. W i n d s o r, Use of a Data Dictionary in a Data Base Environment. The Computer Bulletin, March 1979, s. 24-25.

[32] J. W o j d y ł a, Interaktive Analyse und Projektierung von Dateien in Informationssystemen, [w:] Dialog człowieka z maszyną, Materiały na międzynarodową konferencję naukową, Politechnika Wrocławska,1981, s. 62-71.

[33] J. W o j d y ł a, Kompresja danych w systemach informatycznych. Raport badawczy Instytutu Informatyki II-2-82, Akademia Ekonomiczna we Wrocławiu,1982.

[34] J. W o j d y ł a, Modele wspomaganych komputerem procesów tworzenia systemów informatycznych, [w:] Informatyka. Rachunkowość, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 181, Wrocław 1981, s. 59-69.

[35] J. W o j d y ł a, Organizacja i funkcjonowanie zbiorów danych utworzonych za pomocą wirtualnej metody dostępu /VSAM/. Raport badawczy Instytutu Informatyki, II-1-82, Akademia Ekonomiczna we Wrocławiu,1982.

[36] J. W o j d y ł a, Wspomagane komputerem procesy projektowania i weryfikacji systemów informatycznych, [w:] Materiały konferencji "Współczesne tendencje tworzenia i upowszechniania systemów informatycznych zarządzania", Akademia Ekonomiczna i TNOiK,Katowice 9-11 marca 1981.

[37] S.B. Y a o, K.S. D a s, T.J. T e o r e y, A Dynamic Database Reorganization Algorithm. The University of Michigan, Technical Report 76 DE 2, November 1975 /patrz także, ACM TODS, 1976, nr 2, s. 159-174/.

[38] S. Z i m n o c h o, Zastosowanie interpolacji do przeszukiwania zbiorów danych /w niniejszym tomie/.

THE INTERACTIVE LABORATORY OF THE COMPUTER-AIDED DESIGNING OF DATA BASES

Summary

Some current research trends and results in the field of computer-aided design of data bases are discussed in the article. In this light, the method of three-level data projection, meant for such systems, is presented. These levels are the conceptual, the logical, and the physical one. The necessity of analysing data at any level on the plane of the types of objects as well as analysing occurence is also explained here. The author presents the principles of both structure construction and contents composition of a metabase, i.e., the data base containing the data about the given informatic system. Researche and implementation carried out by the Interactive Laboratory of the Computer-Aided Designing of Data Bases are presented in the paper. Future directions of the Laboratory's researche are also mentioned here.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Mieczysław O w o c, Jolanta S o w i ń s k a
Akademia Ekonomiczna we Wrocławiu

PROGRAMOWANIE PROCESU ŁADOWANIA W SIECIOWYCH BAZACH DANYCH

1. Wstęp

Użytkownik technologii bazy danych napotyka pierwsze trudności w momencie oprogramowywania procesu ładowania bazy. Używane w artykule określenie "proces ładowania bazy danych" będzie dotyczyć zarówno właściwego programu powodującego umieszczenie danych w bazie, jak i niezbędne elementy programowania, które służą do przygotowania danych. Podstawowe trudności napotykane w tym procesie, wynikają w głównej mierze z dużej swobody administratora w zakresie konstruowania struktur logicznych i fizycznych danych. Bardzo ogólne są sugestie producentów systemów baz danych na temat procesu ładowania baz danych. Ponadto osiągnięte w kraju doświadczenia w zakresie omawianego procesu są skromne i nie rozpowszechnione wśród użytkowników systemów baz danych.

Powyższe uwagi są szczególnie istotne w odniesieniu do systemów baz danych wykorzystujących sieciowy model danych. Dopuszczalna w takim modelu struktura wewnątrzzapisowa, a przede wszystkim możliwe związki międzyzapisowe powodują, że tylko w wyniku zastosowania wyszukanych operacji redukcji modelu sieciowego można utworzyć hierarchię danych konieczną do załadowania.

Ostatnią determinantę, mającą wpływ na omawiany proces, stanowi sposób załadowania danych do bazy. Przeciętny użytkownik bazy danych korzysta z czytnika kart perforowanych. Jednakże niektórzy producenci oprogramowania, ze względów sprawnościowych[1], ograniczają proces ładowania bazy do postaci wsadowej.

[1] Zob. [3,4].

Celem artykułu jest scharakteryzowanie programowania procesu ładowania bazy. Rozważania należy traktować jako uwagi do szeroko pojętego procesu wspomagania komputerem konstruowania struktury logicznej i fizycznej bazy. Poświęcono temu ostatnią część artykułu, w której przedstawiono propozycję standaryzacji wybranych etapów omawianego procesu oprogramowywania. W opracowaniu wykorzystano doświadczenia zdobyte podczas badań dotyczących wykorzystywania technologii baz danych w problematyce zatrudnieniowo-płacowej. Prace badawcze prowadzono w latach 1980-1981 w Zakładach Naukowo-Badawczych Akademii Ekonomicznej we Wrocławiu. Posługiwano się wówczas Uniwersalnym Systemem Zarządzania Bazą Danych RODAN. Ładowanie bazy odbywało się na standardowej konfiguracji komputera R-32 wyposażonego w pamięć operacyjną 512 K bajtów oraz podstawowy zestaw urządzeń wejścia i wyjścia i typową konfigurację pamięci zewnętrznych.

2. Klasyfikacja funkcji realizowanych w procesie ładowania bazy danych

O strukturze działań składających się na proces ładowania bazy decydują określone deklaracje w zakresie struktury logicznej, fizycznej oraz podstruktur logicznych. Zależnie od stopnia złożoności tych struktur różne będą funkcje występujące w omawianym procesie. Nie wnikając bliżej w charakterystykę wymienionych wyżej struktur, przedstawiono w niniejszym punkcie pełną listę funkcji omawianego procesu.

Cały proces ładowania bazy można podzielić na etapy, w ramach których są realizowane następujące funkcje:

1/ inicjacja wejścia,
2/ kontrola pierwotna wejścia,
3/ porządkowanie danych,
4/ kompletowanie danych,
5/ kontrola ostateczna wejścia,
6/ ładowanie bazy.

Ze względu na dość zróżnicowany przebieg procesu ładowania bazy wymienione funkcje mogą być realizowane w różnym stopniu. Część z nich - porządkowanie danych, kontrola pierwotna wejścia - ma charakter opcjonalny. Inne, np. inicjacja wejścia czy ładowanie bazy, występują w każdym z procesów szeroko pojętego ładowania bazy.

Pierwsza z funkcji - inicjacja wejścia - jest rozumiana jako etap, w którym następuje przygotowanie danych oraz ich zapisanie na maszynowych nośnikach informacji. W tym etapie istotne znaczenie ma przygotowanie odpowiedniej dokumentacji źródłowej, która decyduje zwykle o jakości danych wprowadzanych do bazy. W przypadku stosowania technologii baz danych jakość danych wejściowych decyduje o poprawności funkcjonowania całej bazy. Wynika to z minimalnej powtarzalności danych w bazie, z możliwości tworzenia związków między danymi w zapisie, a także konfrontowania wartości danych z różnych zapisów.

Celem drugiej funkcji - kontroli pierwotnej wejścia - jest wyeliminowanie tych danych wejściowych, których wartość już w pierwotnym badaniu odbiega od dopuszczalnych wartości wynikających z określonych deklaracji w strukturze i podstrukturach logicznych. Zrealizowanie tej funkcji warunkuje przebieg następnych funkcji. Sprowadza się zatem do badania wartości pól kluczowych i innych mających znaczenie w początkowym okresie ładowania bazy.

Realizacja trzeciej funkcji - porządkowanie danych - polega na usystematyzowaniu danych, zapisanych na maszynowych nośnikach informacji, w sekwencji odpowiedniej dla dalszej obróbki programowej danych. Takie porządkowanie jest konieczne w dwu przypadkach. Po pierwsze wtedy, gdy zapis bazy danych jest kompletowany z mniejszych elementów. Po drugie, gdy konieczne jest zapewnienie sekwencji zapisów dostosowanej do strategii ładowania bazy. Może ona zatem wynikać z powiązań zapisów w zbiory, rodzaju członkostwa w zbiorach itp.

Bardzo istotne jest realizowanie kolejnej funkcji, kompletowania danych. Zapewnia ona bowiem otrzymanie zapisów o strukturze odpowiadającej deklaracjom zawartym w strukturze logicznej i podstrukturach logicznych. Przeprowadzana jest zatem konwersja danych, a także konstruowane są odpowiednie typy zapisów. Ostatnia z czynności ma polegać na tworzeniu zapisów o zmiennej lub stałej długości.

Piątą z wymienionych funkcji - kontrolę ostateczną wejścia - wygodnie jest rozpatrywać w kontekście poprzedniej. Stanowi ona bowiem naturalne uzupełnienie procesu tworzenia zapisów do załadowania, polegające na ich ostatecznej weryfikacji przed zapisaniem w bazie. Podobnie jak inne, typowe funkcje kontrol-

ne prowadzi ona do skonfrontowania wartości danych z dopuszczalnym zakresem, kompletności danych w zapisie itp. Zadania te, oprócz trybu programowego, powinny być wykonywane także w sposób wizualny.

Wreszcie realizacją ostatniej funkcji, ładowania bazy, prowadzi do umieszczenia danych w bazie zgodnie z parametrami przyjętymi w logicznej i fizycznej strukturze. Wymaga ona opracowania odpowiedniej strategii umieszczania zapisów w bazie, przy czym jej przebieg może się odbywać w kilku fazach. Decydują o tym charakterystyki zbiorów występujących w bazie /proces ładowania rozpoczyna się od zapisów nadrzędnych, a kończy na zapisach podrzędnych/. W razie nieskomplikowanej struktury logicznej bazy mamy do czynienia z procesem jednofazowym. W innych, bardziej złożonych przypadkach konieczne jest przygotowanie bazy do załadowania w kilku fazach.

Z pobieżnego przeglądu funkcji, składających się na przebieg procesu ładowania, wynika, że nie jest to proces poddający się łatwo standaryzacji. Administrator i użytkownik bazy stają bowiem przed wieloma niewiadomymi, które można poprawnie rozwiązać metodą prób i błędów bądź też opracować określony model procesu ładowania. Każda z wymienionych funkcji podlegających oprogramowaniu ma dość zróżnicowaną strukturę wewnętrzną, czas oprogramowania itp. Prawidłowości występujące podczas programowania omawianych funkcji przedstawiono w kolejnych partiach artykułu.

3. Charakterystyka programów załadowujących bazę danych

Programy załadowujące bazę danych stanowią jeden z ważniejszych etapów w procesie funkcjonowania bazy. Podczas realizacji takiego programu następuje wiązanie /zadeklarowanych uprzednio w strukturze logicznej i fizycznej/ charakterystyk zapisów oraz związków pomiędzy nimi z konkretnymi wartościami. Powstaje wówczas układ informacji, który może być zarządzany przez odpowiednie oprogramowanie systemowe. Jednocześnie załadowanie decyduje o poprawności bazy w sensie logicznym i fizycznym. Z tego względu należałoby zwrócić uwagę na szczególne cechy tego typu oprogramowania.

Należy pamiętać o trzypoziomowym odwzorowaniu danych; od struktury logicznej /schemat bazy/ przez strukturę fizyczną do podstruktur logicznych /podschematy bazy/. W każdym z tych odwzorowań następuje określenie parametrów wpływających na konstrukcję programów ładujących. Stosownie do wymienionych odwzorowań zostaną wytypowane parametry wpływające na proces ładowania. Kolejno będą one dotyczyć:

a/ deklaracji używanych w schemacie,

b/ planowanego przydziału pamięci,

c/ postaci podschematów.

W ramach pierwszej grupy należy zwrócić uwagę na deklaracje: określające typ danej, charakter danej, miejsce w strukturze zapisu, klauzule tajności, metody kodowania. Dość jednoznacznie wyznaczają one strukturę zapisów ładowanych do bazy. Istotne są również deklaracje związane z tworzeniem struktur międzyzapisowych, spośród których należy wymienić te, które definiują typ zapisu, porządek wewnętrzny zapisów w zbiorze czy sposób umieszczania i wyszukiwania zapisów w bazie. Struktura logiczna sformułowana w schemacie bazy musi być uwzględniona przy konstruowaniu programów ładujących bazę. Wyznacza ona część deklaratywną zapisów umieszczanych w bazie, a także określa sekwencje procedur realizowanych w trakcie ładowania kolejnych zapisów.

Druga grupa parametrów bazy, mających wpływ na przebieg procesu ładowania, dotyczy struktury fizycznej bazy. Spośród ważniejszych deklaracji dotyczących struktury fizycznej najistotniejsze wydają się parametry obszarów bazy, sposób łączenia zapisów w zbiory, a także niektóre deklaracje bezpośrednio związane z procesem ładowania, np. pozostawienie wolnych obszarów w bazie. Wprawdzie wymienione parametry nie wpływają bezpośrednio na konstrukcje programów ładujących, lecz z pewnością decydują o czasie trwania ładowania bazy. Z tego też powodu powinny być one uwzględnione podczas wybierania różnych warunków struktury fizycznej, właśnie w aspekcie przyszłego ładowania bazy.

Ostatnia grupa parametrów - dotycząca deklaracji w zakresie podstruktur logicznych - również bezpośrednio wpływa na konstrukcje programów ładujących. Dopuszczalne bowiem w podschematach bazy redefinicje bazy danych zezwalają zarówno na mo-

dyfikacje struktur wewnątrzzapisowych, jak i związków między zapisami. Wpływają one zatem na część deklaracyjną programu ładującego /możliwość pominięcia określonych zapisów czy danych/ i część proceduralną wyrażającą kolejne operacje na zapisach bazy.

Przedstawione zewnętrzne uwarunkowania programów ładujących bazę stanowią jeden z elementów ich charakterystyki. Spośród innych cech tych programów istotne znaczenie mają:

a/ programy ładujące, stanowiące "mieszankę" komend w języku bazowym oraz w Języku Manipulowania Danymi,

b/ programy przetwarzane przez tzw. preprocesor,

c/ programy mające przesyłaną z zewnątrz do programu część deklaracyjną, wynikającą ze schematów i podschematów,

d/ część proceduralna programów składająca się z komend: przygotowujących bazę do pracy, zapisujących dane do bazy, tworzących zbiory z umieszczonych w bazie zapisów, wyszukujących zapisy nadrzędne, zamykających bazę,

e/ diagnostyka obsługi błędów i przerwań w programach, która pozwala ocenić stan załadowania bazy oraz determinuje dalsze operacje na bazie.

Przede wszystkim dwie ostatnie cechy programów ładujących różnią je od innych programów działających na bazie. Przeważają w nich bowiem komendy wykonywania "organicznych" zmian w bazie oraz specyficzna diagnostyka charakteryzująca stan bazy.

4. Programowanie procesu załadowania bazy danych

Załadowanie bazy danych jest procesem złożonym, składającym się zwykle z kilku etapów. W ramach struktury tego procesu występują funkcje, wymienione poprzednio, których programowanie może być rozwiązywane w różny sposób. Przede wszystkim dla konkretnej bazy niektóre z tych funkcji mogą nie wystąpić, w innym zaś przypadku może nastąpić programowa integracja funkcji pochodnych. Rozważmy zatem, jakie są programowe uwarunkowania realizacji omawianego procesu.

Pierwszy z etapów obejmuje, jak już sygnalizowano, programy wstępnego utworzenia zapisów. W ramach tych programów dokonuje się zazwyczaj wstępnej kontroli wejścia. Możliwość realizacji pierwszych dwu funkcji w jednym programie wynika z prostoty kon-

strukcyjnej programów oraz z udogodnień, na jakie pozwalają poszczególne języki programowania[2]. Podlegają one dość wszechstronnej kontroli formalnej poszczególnych pól zapisów, kontroli logicznej dokonywanej w odniesieniu do tych pól, jak i różnych zapisów wynikających z zależności między polami.

Jednym z podstawowych parametrów bazy, decydujących o konstrukcji programów realizujących proces załadowania, jest wielkość i struktura wewnętrzna zapisów ładowania do bazy, a także zagadnienie tzw. uporządkowania zapisów w ramach zbioru[3]. Wartością graniczną jest długość zapisu przekraczająca 80 znaków i wymagająca w tej sytuacji oprogramowania dalszych funkcji. Kolejną grupą programów, przy obrabianiu zapisów o długości przekraczającej 80 znaków, są programy sortujące.

Sortuje się zapisy fizyczne do 80 znaków, stanowiące części składowe przyszłych logicznych zapisów umieszczonych w bazie. Sortowanie to można realizować jednym z programów sortujących OS/JS, np. IERRC000, lub wywołaniem tego programu /używając procedury CALL/ do programu własnego realizującego np. funkcje inicjacji wejścia, kontroli wstępnej wejścia bądź też funkcji kompletacji danych. Atrybuty sortowania deklarowane w programach sortujących wynikają z konieczności dalszej kompletacji danych w zapisy bądź też są podyktowane strategią ładowania bazy.

Następna grupa programów to programy kompletujące, których funkcją jest "sklejanie" przetwarzanych zapisów w całość logiczną i formatowanie zapisów zgodnie z charakterystyką zdefiniowaną w schemacie bazy danych. Jedynym kryterium kompletacji danych jest wartość pola kluczowego, która stanowi punkt odniesienia dla określonej grupy danych. Programy kompletujące /ze względu na bliskość deklaratywną danych podlegających kompletacji i zamieszczanych w bazie/ muszą być pisane w języku bazowym, dopuszczalnym dla danego systemu zarządzania bazą danych. W wyniku sukcesywnej realizacji wymienionych poprzednio etapów uzyskuje się zbiory zapisów o charakterystyce przygotowanej do wiązania tych zapisów z deklaracjami sformułowanymi w odpowiednim podschemacie czy schemacie bazy danych.

---

[2] Zob. [5].

[3] A więc rola, jaką pełni zapis w zbiorze /właściciel lub składowy/, oraz tryb wiązania tych zapisów w zbiorze.

Ostatnią grupę programów realizujących proces załadowania bazy należy uznać za najważniejszą w całym procesie. Funkcją programów ładujących jest bowiem odwzorowanie przygotowanych uprzednio zapisów w strukturę fizyczną bazy z uwzględnieniem deklaracji wynikających ze struktury logicznej. Programy zaliczone do tej grupy obejmują komendy Języka Manipulacji Danymi, obsługującego wszystkie odwołania do bazy, a także komendy języka bazowego, które obsługują odwołania do zbiorów zapisów umieszczanych w bazie.

Oto typowy układ działań, charakterystyczny dla programów ładujących:

a/ wywołanie podschematu bazy danych "zorientowanego" na proces ładowania,

b/ otwarcie obszarów bazy,

c/ wprowadzanie zapisów do bazy,

d/ wyszukiwanie zapisów stanowiących "wierzchołek" zbioru dla zapisu podrzędnego,

e/ sprawdzenie prawdziwości warunku związanego z kwalifikacją zapisu przedmiotowego do zbioru,

f/ zamykanie obszarów bazy danych.

W programach ładujących działania wymienione w punktach c, d, e mają charakter iteracyjny. Istotna jest kolejność wykonywania tych działań.

O kolejności ładowania zapisów do bazy decyduje klauzula określająca tryb lokowania w schemacie bazy danych. Należy ustalić hierarchię zapisów bazy, uwzględniającą zarówno tryby lokowania poszczególnych zapisów, jak i występujące między nimi zależności logiczne. Ładowanie powinny zatem rozpocząć zapisy lokowane metodą[4] CALC i te, które nie były definiowane w żadnym typie zbioru jako automatyczne zapisy członkowskie. Kolejną grupę zapisów przeznaczonych do umieszczenia w bazie stanowią zapisy lokowane tzw. metodą SYSTEM oraz metodą CALC, uczestniczące w zbiorach jako tzw. automatyczne zapisy członkowskie. W tym momencie szczególnie ważne jest grupowe umieszczanie tych zapisów zaliczonych do jednego zbioru, co wydatnie skraca czas ładowania bazy. W końcowym etapie ładowania umieszcza się w bazie zapisy lokowane metodą VIA.

---

[4] Szczegółowe uwagi na temat metod lokowania zawiera dokumentacja systemu RODAN [6].

Uwagi na temat przebiegu procesu ładowania nasunęły się autorom po przeprowadzeniu badań. Uwagi uzupełniono bardziej szczegółową charakterystyką bazy danych, stanowiącej przedmiot badań, programami opracowanymi dla obsługi tego procesu, a także bezpośrednimi wnioskami wynikającymi ze związków między poszczególnymi definicjami struktury i podstruktur logicznych oraz fizycznych bazy. Prezentują je tab. 1 i 2 oraz rys. 1.

Jak wynika z tab. 1, ładowana baza danych jest różnorodna i w pewnym stopniu skomplikowana. Baza ta zawiera czternaście typów zapisów, z których sześć - w zależności od wystąpienia w zbiorze danych - pełni bądź rolę zapisu nadrzędnego, bądź rolę zapisu podrzędnego. Pozostałe zapisy są podrzędne. Zapisy o symbolach 1, 2, 4, 5, 6, 8, 11, 12, 13, 14 są także zapisami podrzędnymi w zbiorach systemowych, gdzie zapisem nadrzędnym jest system zarządzania bazą danych. Zapisy mają różną długość, sześć typów zapisów ma zmienną długość. Struktura fizyczna bazy jest podzielona na siedem obszarów, których podstawą wyodrębnienia był temat ładowanych zapisów. Podczas ładowania bazy posłużono się dwiema metodami lokowania: CALC i VIA.

Na rys. 1 zapisy prezentowano za pomocą kółek, w których wpisano symbol zapisu podany w tab. 1. Zależności między zapisami oznaczono prostymi, zakończonymi strzałkami. Zapis, z którego wychodzi prosta, jest zapisem nadrzędnym, a strzałka wskazuje zapis podrzędny. Jeden zapis nadrzędny ma tyle zapisów podrzędnych, ile zawiera prostych, zakończonych strzałkami. Niektóre zapisy są zarówno nadrzędnymi, jak i podrzędnymi. Rolę zapisu nadrzędnego w niektórych przypadkach pełni system zarządzania bazą danych. Rysunek obrazuje także przyjętą strategię procesu ładowania bazy danych. Wyodrębniono poszczególne etapy realizacji tego procesu i nazwano je: LADA 1, LADA 2, LADA 3. Kolejność realizowania etapów jest bardzo istotna i nie można jej zmieniać. Przy wyborze przedstawionej strategii kierowano się następującymi zasadami:

1/ w pierwszej kolejności ładowano zapisy bazy danych pełniące tylko funkcję zapisów nadrzędnych i ich zapisy podrzędne nie będące równocześnie nadrzędnymi,

2/ w następnej kolejności uwzględniono zapisy, które pełnią rolę i nadrzędnych, i podrzędnych,

Tabela 1

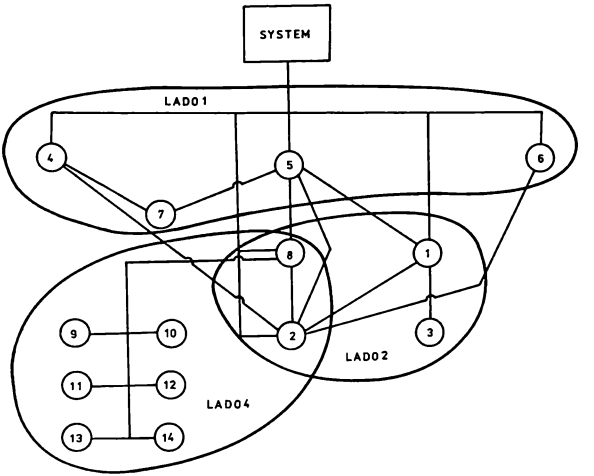
Charakterystyka ładowanej bazy danych

Nazwa zapisu	Symbol zapisu	Zapis		Metoda lokowania	Klucz	Nazwa obszaru	Wystąpienia w zbiorach	
		typ	długość w bajtach				liczba zbiorów	rola zapisu w zbiorach
Dane personalne	1	zmienny	507-944	CALC	NUMER	PRACOWNIK1	3	N/1/,p/2/
Kwalifikacje	2	stały	515	CALC	NUMER	PRACOWNIK2	8	N/1/,p/7/
Dane socjalne	3	stały	150	CALC	NUMER	PRACOWNIK3	1	p/1/
Stanowisko-zawód	4	zmienny	35-275	CALC	STANZAW	STANOWISKO	3	N/2/,p/1/
Jednostka organizacyjna	5	stały	94	CALC	JEDNORG	JEDNOSTKA	7	N/5/,p/2/
Tabele zaszeregowania	6	stały	65	CALC	RODZTAB	PRACOWNIK2	2	N/1/,p/1/
Stanowisko-zawód w jednostce organizacyjnej	7	stały	33	VIA		RELACJA	2	p/2/
Zarobki	8	stały	191	CALC	NUMER	PLACA	4	N/2/,p/2/
Pożyczki	9	stały	63	VIA		PLACA	1	p/1/
Zmienne dopłaty	10	zmienny	14-64	VIA		PLACA	1	p/1/
Karty pracy	11	zmienny	32-140	VIA		PLACA	2	p/2/
Godziny dodatkowe	12	stały	47	VIA		PLACA	2	p/2/
Urlopy	13	zmienny	64-204	VIA		PLACA	2	p/2/
Czas nieprzepracowany	14	zmienny	59-139	VIA		PLACA	2	p/2/

N - zapis nadrzędny; p - zapis podrzędny; / / - liczba zbiorów, w których zapis uczestniczy.

3/ w etapie ostatnim włączono zapisy, które są zapisami podrzędnymi.

Dla każdego etapu procesu ładowania bazy danych opracowano odrębną podstrukturę logiczną w postaci odpowiednich podschematów.



Rys. 1. Hierarchia zapisów bazy

W tabeli 2 scharakteryzowano programy realizujące poszczególne etapy procesu ładowania bazy. Umieszczono w niej cztery typy programów i dla każdego z nich podano średnie czasy kompilacji i wykonania oraz wykorzystanie pamięci operacyjnej. Z tabeli wynika, że najbardziej czasochłonnym oraz pamięciochłonnym etapem był program ładujący. Jest to zrozumiałe ze względu na

bardzo złożone operacje, jakie wykonuje system w momencie wiązania opisu struktury logicznej z danymi bazy. Należy zwrócić uwagę, że w przypadku pamięci operacyjnej 512 K bajtów ładowanie bazy danych, będącej przedmiotem eksperymentu, odbywało się na granicy możliwości realizacyjnych systemu.

W ramach programów stanowiących etap przygotowania danych do zapisania w bazie /programy CZYT, SORT, LACZ/ przetworzono zapisy stanowiące struktury danych o charakterystyce coraz bardziej zbliżonej do postaci zadeklarowanych w schemacie. Z tego też względu różna jest liczba zapisów przetwarzanych w poszczególnych programach. Kryterium podziału poszczególnych etapów programowania na programy była jednolitość tematyczno-identyfikacyjna poszczególnych zapisów, np. łączenie zapisów dotyczących charakterystyki pracownika w programie LACZ 3 czy sortowanie zapisów zawierających pole kluczowe-numer ewidencyjny w programie SORT 1.

Z analizy parametrów eksploatacyjnych grupy programów przygotowujących dane do załadowania wynika, że drugą /po programach ładujących/ grupą sprawiającą trudności były programy łączące. Zródłem tego było realizowanie kilku funkcji przez te programy. Należy zwrócić uwagę, że ze względu na złożoność tych programów postanowiono je zmodyfikować. Z dwu spośród nich /LACZ 1 oraz LACZ 4/ utworzono grupę czterech dodatkowych programów. Pozostałe programy typu czytająco-sortującego nie sprawiały autorom trudności. Ich czasochłonność i pamięciochłonność nie odbiegały od oczekiwanych. Wszystkie programy przygotowano w języku PL/1. W programowaniu trwającym dwa miesiące, brały udział trzy osoby.

O przebiegu procesu ładowania decydowały dwa parametry: wielkość i złożoność bazy. Na złożoność bazy składa się liczba i jakość deklaracji zawartych w schemacie i podschematach bazy danych. Poszczególne deklaracje w różnym stopniu wpływają na realizację poszczególnych komend JMD w programie ładującym bazę. Mogą one dotyczyć zarówno czasów realizacji odpowiednich komend, jak i pamięciochłonności wymaganej przez program ładujący. Ze względu na trudność wyodrębnienia tych związków w czystej postaci przyjęto, że mają one charakter łączny, czyli dotyczą zajętości pamięci operacyjnej oraz czasochłonności przetwarzania.

Tabela 2

Charakterystyka programów realizujących proces ładowania eksperymentalnej bazy danych

Lp.	Nazwa programu	Liczba przetwarzanych zapisów	Funkcja programu	Średni czas w min.		Zajętość PAO w kbajtach
				kompilacji	wykonania	
1	CZYT1 CZYT2 CZYT3	120 185 565	wczytanie danych, wstępna kontrola, zapisanie danych	10	15	50
2	SORT1 SORT2 SORT3 SORT4	40 75 60 565	sortowanie zapisów	1	3	40
3	LACZ1 LACZ2 LACZ3 LACZ4	40 75 15 205	kontrola logiczna pól w zapisach, łączenie zapisów, formatowanie	15	25	55
4	LADO1 LADO2 LADO3	80 160 175	ładowanie bazy danych	50	20	250

Wymienione uwarunkowania uwzględniono podczas charakteryzowania związków między deklaracjami struktur logicznych i fizycznych oraz komend JMD, które występują zwykle w programach ładujących. Najwięcej trudności podczas charakteryzowania omawianych związków sprawiało określenie siły związku między sygnalizowanymi deklaracjami a realizacją komend JMD. Zdecydowano się na skalę dwustopniową i w przypadku stwierdzenia oczywistych związków określono je mianem znaczących /symbol 1/, w pozostałych natomiast przypadkach określono je jako nieznaczące /symbol 0/. Wyniki analizy omawianych związków zawiera tabela 3.

Z przedstawionej charakterystyki można wyciągnąć dwie grupy wniosków. Pierwsza z nich dotyczy "odporności"[5] danej grupy ko-

[5] "Odporność" danej komendy na deklaracje struktur należy rozumieć jako niezależność jej wykonania od szczegółowych parametrów bazy mogących wpłynąć na to wykonanie.

mend na deklaracje w schemacie i podschematach /kolumny 3-5 w tabeli 3/. Okazuje się, że najbardziej niezależnymi komendami są te, które przygotowują bazę do pracy oraz kończą przetwarza-

Tabela 3

Charakterystyka związków pomiędzy deklaracjami struktur bazy oraz wybranymi komendami JMD

Lp.	Komendy JMD / Deklaracje struktur	Przygotowawczo-zakończeniowe: INVOKE OPEN CLOSE	Ładujące dane do bazy STORE INSERT	Wyszukiwawcze Formaty FIND
1	2	3	4	5
1	SCHEMA	1	0	0
SL	ON,PRIVACY/schemat/	1	0	0
	RECORD	0	1	1
	LOCATION,WITHIN	0	1	1
	ON,PRIVACY/rekord/	0	1	1
	dana-bazy	0	1	1
	PICTURE,TYPE,OCCURS	0	1	1
	RESULT,SOURCE	0	1	1
	CHECK,ENCODING	0	1	0
	ON,PRIVACY/dana/	0	1	1
	SET	0	1	1
	OWNER	0	1	1
	ORDER	0	1	1
	ON,PRIVACY/zbiór/	0	1	1
	MEMBER	0	1	1
	KEY,SEARCH	0	1	1
	SELECTION	0	1	1
2	AREA	1	1	1
SF	ALLOCATE	1	1	0
	LOAD,CALC	1	1	0
	RECORD	1	1	0
	LENGTH	1	1	0
	PLACE,SCAN	1	1	0
	SET	1	1	1
	MODE	1	1	1
	MEMBER	1	1	1
	INDEX	1	1	1
3	SUBSCHEMA	1	1	1
PSL	RENAME,COPY	0	1	1
	RECORD	0	1	1
	SET	0	1	1

Skala ocen związków: 0 - nieznaczący; 1 - znaczący; SL - struktura logiczna, SF - struktura fizyczna; PSL - podstruktura logiczna.

nie /komendy INVOKE, OPEN, CLOSE/. Ze zrozumiałych względów najbardziej "wrażliwymi"[6] na deklaracje struktur logicznych i fizycznych są komendy bezpośrednio związane z ładowaniem danych do bazy /komendy STORE, INSERT/. Przyjętą w artykule miarą "wrażliwości" czy "odporności" była suma ocen poszczególnych związków. Wnioski można również formułować z punktu widzenia wpływu danej deklaracji struktury na przebieg wykonywania komendy JMD. Daje to odpowiedź na pytanie, które z elementów opisu struktur bazy należy rozważać w kontekście opracowywania programów ładujących. Z tabeli 3 wynika, że podstawowe znaczenie mają deklaracje dotyczące struktury logicznej /zdania opisujące zapisy i zbiory/, a także niektóre parametry struktury fizycznej.

Wprawdzie wnioski, jakie można sformułować na podstawie tabeli 3, nie odbiegają od oczekiwanych, jednak bez wątpienia stanowią uzasadnienie niektórych rozwiązań konstrukcji programów ładujących bazę.

5. Możliwości standaryzacyjne programowania procesu załadowania bazy

Jak wiadomo, oferty producentów oprogramowania systemów baz danych w zakresie usprawniania procesu ładowania bazy są różnorodne, ale skromne. Przyczyną takiego stanu /jak już wcześniej sygnalizowano/ jest złożoność omawianego procesu, która sprawia, że algorytmizowanie i standaryzowanie czynności występujących w procesie ładowania bazy jest dziś trudne. Wydaje się jednak, że przynajmniej w pewnym zakresie można mówić o pewnych, zrutynizowanych czynnościach wykonywanych podczas ładowania bazy. Za elementy standaryzacji omawianego procesu można wstępnie uznać:

a/ generowanie struktur zapisów bazy istotnych dla programów poprzedzających ładowanie bazy,

b/ generowanie hierarchii zapisów bazy warunkującej sposób jej ładowania.

---

[6] "Wrażliwość" jest rozumiana jako przeciwieństwo cechy odporności i polega na bezpośrednim wpływie określonej deklaracji struktury na czasochłonność lub pamięciochłonność wykonywanej komendy JMD.

Przedstawimy założenia programowe dotyczące funkcji, które wynikają z przedstawionej propozycji standaryzacji. Każda z funkcji może stanowić odrębne rozwiązanie programowe, przy czym można uznać, że są to funkcje wzajemnie się uzupełniające.

Pierwsza funkcja - generowanie struktur zapisów - ma szczególne znaczenie w tworzeniu struktur logicznych, charakteryzujących się dużą złożonością wewnątrzzapisową, rozumianą jako tworzenie struktur zapisów o wielu poziomach hierarchii, używanie grup danych powtarzalnych, konstruowanie zapisów o zmiennej długości itp. W programach przygotowujących ładowanie bazy wymaga się formy zapisów zbliżonej do deklaracji w schemacie bazy danych. Istotna jest zatem możliwość wygenerowania poszczególnych pól zapisu o charakterystyce nakreślonej w schemacie bazy. Program realizujący taką funkcję generacyjną musi być napisany /ze względu na zachodzące w programie konwersje/ w języku bazowym, akceptowanym przez konkretny system bazy danych. Działanie takiego programu pozwoliłoby na wygenerowanie, na podstawie schematu bazy, struktur zapisów o charakterystyce żądanej przez użytkownika. Możliwe byłoby wówczas przesłanie części deklaracyjnej do poszczególnych programów przygotowujących załadowanie bazy. Dzięki temu można by, w przypadku złożonych struktur wewnątrzzapisowych, uniknąć niepowodzeń w procesie ładowania.

Druga funkcja - generowanie opcji kontrolnych - stanowi naturalne poszerzenie części deklaratywnej o możliwe opcje kontrolne. Można je uznać za kontrolę formalną autonomicznie dokonywaną przez program na podstawie deklaracji zawartych w schemacie. Ze względu jednak na konieczność zachowania integralności danych, omawiany aspekt standaryzacyjny oprogramowania wymaga przeprowadzania badań analitycznych dotyczących związków między polami różnych typów zapisów. Spośród możliwych operacji dokonywanych na danych należałoby wymienić: kontrolę formalną, kontrolę grup powtarzanych w zapisach zmiennej długości, a także kontrolę unikalności określonych wartości danych w bazie. Ostatnia z możliwych opcji wymagałaby przeglądania wielu zapisów przygotowanych do ładowania oraz analizowania struktur międzyzapisowych w bazie.

Generowanie hierarchii zapisów proponowane do standaryzacji funkcji wydaje się najbardziej atrakcyjne w sensie badawczym.

Należałoby bowiem analizować powiązania między zapisami, by na ich podstawie wygenerować hierarchię zapisów od najwyżej do najniżej położonych w strukturze. Oczywiście istnieją ograniczone możliwości zbadania wszystkich, możliwych powiązań między zapisami, ale w uproszczeniu zakres badań programu byłby równy zdolnościom analitycznym systemu bazy danych. Możliwe wydaje się natomiast wygenerowanie zależności dwustronnych /między różnymi zapisami/, co sprowadziłoby się do badania uczestnictwa zapisów w zbiorach bazy danych.

Ze względu na potrzeby użytkowników zasygnalizowane założenia programowe powinny stanowić zalążek rozbudowy oprogramowania wspomagającego procesy zakładania bazy danych przy jednoczesnej automatyzacji procesu projektowania systemów informatycznych. Problem ten już dość wszechstronnie rozwiązano w przypadku niektórych hierarchicznych baz danych, zwłaszcza systemu IMS[7]. Brakuje natomiast takiego oprogramowania dla dostępnego w kraju systemu bazy danych RODAN, wykorzystującego sieciowy model danych.

Zatem wydaje się, że proponowane do oprogramowania funkcje mogą stać się jednym z istotnych uzupełnień funkcjonującego systemu RODAN.

Zawarte w artykule rozważania, dotyczące oprogramowania procesu ładowania bazy, można ująć w postaci wniosków:

1/ proces ładowania bazy w systemach wykorzystujących model sieciowy jest procesem złożonym, wymagającym od programisty dobrej znajomości zastosowań - funkcjonowania systemu baz danych,

2/ obecny etap rozwoju systemów, nie posiadających usprawnień w zakresie ładowania baz danych, wymaga wzbogacenia tych systemów o oprogramowanie wspierające realizację procesu ładowania bazy,

3/ za jeden z kierunków rozwoju systemów baz danych, w aspekcie problemu ładowania bazy, można uznać oprogramowanie funkcji generacyjnych dotyczących: struktur danych, opcji kontrolnych oraz hierarchii zapisów.

---

[7] Patrz 2, s. 59-64 .

LITERATURA

[1] C.J. D a t e, Wprowadzenie do systemów baz danych, Warszawa 1981, WNT.

[2] IBM System/370 and Processors Bibliography, IBM 1979, GC20-0001-3.

[3] Information Management System/360, Version 2. System/Application Design Guide, IBM 1974, SH 20-0910.

[4] Information Managament System/360, Version 2. Utilities Reference Manual, IBM 1974, SH 20-0915.

[5] J.E. N i c h o l l s, Struktura języków programowania, Warszawa 1980, WNT.

[6] Uniwersalny System Zarządzania Bazą Danych RODAN. Dokumentacja użytkowa, cz. I-IV, Warszawa 1979, CPiZI.

THE PROGRAMMING OF LOADING PROCESS IN NETWORK DATA BASES

Summary

The process of data bases loading, due to a highly complicated character of the processed data description, is one of the vital problems of data base systems functioning. The article presents the basic conditions connected with the software of the process discussed. The characteristics of the software for the base loading process have been worked out through using empirical research and concern problems connected with employment and wages, the latter having been carried out by means of the RODAN system. Empirical research has made it possible to formulate some theses on directions of the automatization of the process discussed.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Marek S k w a r n i k
Akademia Ekonomiczna we Wrocławiu

ANALIZA MOŻLIWOSCI WYKORZYSTANIA ZASOBÓW SYSTEMU ZARZADZANIA BAZĄ DANYCH W PROCESIE OCHRONY JEJ JAKOSCI

1. Wstęp

Odpowiednia jakość bazy danych jest jednym z głównych warunków efektywnego jej wykorzystania. Aby osiągnąć i utrzymać wymagany poziom jakości należy wyeliminować zakłócenia. Jest to możliwe dzięki powtarzającym się procesom neutralizacji, z których każdy może oddziaływać na wyróżniony podzbiór zakłóceń występujący w unikalnie zidentyfikowanym zestawie danych. Spójny kompleks procesów tego typu, realizowanych w ramach określonego systemu informatycznego, określono mianem procesu ochrony jakości bazy.

W artykule analizuje się jedną z trzech podstawowych grup. Przedmiotem badań są procesy neutralizacji zakłóceń występujących w danych merytorycznych bazy[1]. Ze względu na częstość występowania tych zakłóceń stanowią one największe zagrożenie jakości bazy. Jednocześnie są one najbardziej zróżnicowane, zwłaszcza z punktu widzenia różnorodności form występowania błędów i przyczyn, które mogą spowodować ich zaistnienie.

W literaturze przedmiotu niewiele miejsca poświęca się problemom neutralizacji zakłóceń merytorycznych. Uwagę skupia się na pozostałych dwóch grupach zakłóceń, które - w przeciwieństwie

[1] Danymi merytorycznymi nazwano dane wprowadzone do bazy z obiektu w celu zaspokojenia potrzeb informacyjnych końcowych użytkowników systemu. Większość spośród nich opisuje stany i zdarzenia zachodzące w obiekcie, którego system dotyczy. Zakłócenia występujące w tych danych będą określane mianem zakłóceń merytorycznych bazy bądź zakłóceń, błędów, zniekształceń i przekłamań. Ostatniego terminu będziemy używać w ograniczonym zakresie, gdyż przekłamanie jest właściwie jedną z kilku podstawowych form ujawnienia się zakłócenia.

do przedmiotu rozważań - można uznać za specyficzne dla baz danych[2]. Zakłócenia danych merytorycznych są zbliżone charakterem i formą do zakłóceń występujących w danych w trakcie innych etapów procesu przetwarzania, niekoniecznie w ramach systemów informatycznych z bazą danych. W pracach poświęconych tym zagadnieniom /problematyka szeroko pojętej jakości danych/ rzadko uwzględnia się specyfikę bazy danych jako ugrupowania danych odmiennie zarządzanego [2, s. 198-234;7;12]. Celem autora jest częściowe uzupełnienie istniejącej luki i dlatego analizuje on możliwości wykorzystania systemu zarządzania bazą do celów neutralizacji.

Uznano, że określenie kompleksowego zbioru funkcjonalnych i technologicznych zadań, które powinny być realizowane w trakcie procesu ochrony jakości bazy, jest koniecznym warunkiem dokonania pełnej oceny. Jedynie porównanie tego zbioru z zasobami dostępnymi w ramach systemu zarządzania umożliwi autorowi postawienie kompleksowej diagnozy. W związku z tym artykuł podzielono na dwie części. W pierwszej z nich starano się szczegółowo zidentyfikować zbiór wariantów realizacji wycinkowych zadań /akcji/ wchodzących w skład procesów neutralizacji zakłóceń merytorycznych. Część druga to opis dostępnych zasobów systemu zarządzania bazą, analiza porównawcza i próba postawienia diagnozy.

2. Model procesu neutralizacji zakłóceń merytorycznych bazy danych

Procesy neutralizacji zakłóceń merytorycznych mogą być realizowane w wielu wariantach. Zróżnicowanie w ramach ich zbioru wynika stąd, że: po pierwsze, przedmiotem ich działania są grupy zakłóceń występujące w ramach odmiennych zestawów danych, po drugie, pewne typy zakłóceń można neutralizować różnymi metodami. Występujące odrębności powodują, że niezmiernie trudne jest skonstruowanie jednolitego szczegółowego modelu neutralizacji zakłóceń uogólniającego zbiór rzeczywiście realizowanych procesów.

[2] Pozostałe grupy zakłóceń jakości bazy to zakłócenia technologiczne /dotyczą danych bazy o charakterze technologicznym, np. łączników czy indeksów/ oraz fizyczne uszkodzenia bazy. Na temat zasad neutralizacji tej ostatniej grupy zniekształceń pisano w [8]; zasady wykrywania i likwidacji błędów technologicznych opisano szczegółowo w pracy [1].

W artykule podjęto próbę zbudowania takiego modelu. Ponieważ niewiele jest eksploatowanych systemów z bazą danych, które mogły być przedmiotem bezpośrednich badań, większość spośród opisanych dalej wariantów neutralizacji zidentyfikowano na podstawie dostępnej literatury [2;4;6;7;11;43].

Wykorzystanie literatury przedmiotu jako podstawowego źródła informacji o dopuszczalnych wariantach neutralizacji błędów jest przedsięwzięciem dyskusyjnym. Zdecydowano się jednak nań nie tylko ze względu na wspomniane ograniczenia. Inną, znacznie bardziej istotną przyczyną takiego postępowania było dążenie do kompleksowej inwentaryzacji wariantów neutralizacji omawianej grupy zakłóceń, w tym również takich, które stosuje się bardzo rzadko lub w wyjątkowych sytuacjach. W jednym z poprzednich artykułów stwierdzono bowiem [5], że działanie zmierzające do ustalenia zbioru wariantów neutralizacji, które będą wykorzystywane do ochrony jakości określonej bazy, ma charakter niealgorytmiczny, a stosowane kryteria są intuicyjne bądź szacunkowe.

Gdy nie ma racjonalnych formalnych algorytmów wyboru /"dobroci" poszczególnych rozwiązań/, a możliwość weryfikacji wybranego zbioru wariantów neutralizacji przez użytkowników jest iluzoryczna, Administrator może dokonywać selekcji kierując się subiektywnymi kryteriami. Jednym z podstawowych, pozytywnych kryteriów /jeśli nie podstawowym/ jest możliwość realizowania wariantu neutralizacji, korzystając w sposób parametryczny z zasobów systemu zarządzania bazą. W tej sytuacji wskazanie wszystkich wariantów, których stosowanie można uzasadnić merytorycznie, jest jedyną metodą ominięcia błędnego koła, istniejącego wówczas, gdy selekcjonuje się opisywane rozwiązania z punktu widzenia częstości ich realizowania.

Wykorzystanie w praktyce wyników analizy działających systemów ochrony jakości bazy komplikuje specyfika zakłóceń merytorycznych. W większości przypadków przyczyną ich powstawania są błędy w danych wejściowych wynikające głównie z nieprawidłowego wypełnienia dokumentów, niewłaściwego przeniesienia na maszynowy nośnik /w przypadku wsadowej aktualizacji bazy/ lub niepoprawnego wprowadzenia transakcji /w trakcie aktualizacji on-line/[3].

[3] W pracy [2] liczbę błędów danych merytorycznych spowodowanych tymi przyczynami szacuje się na 80-90% ogółu.

Należy im zapobiegać już na etapach przetwarzania, poprzedzających ich wprowadzenie do bazy. Z tego powodu liczba procedur neutralizacji zakłóceń bazy wytypowanych dla danego systemu jest ściśle uzależniona od oceny "dobroci" zestawu omawianych środków zapobiegawczych wbudowanych uprzednio w system informatyczny.

Ponieważ jednym z podstawowych kierunków racjonalizacji ochrony przed błędnymi danymi jest wykrywanie ich możliwie najszybciej i najbliżej miejsca powstania /co daje szansę minimalizacji nakładów na usunięcie zniekształceń i ograniczenia skutków ich oddziaływania/, należy tylko wpływać w koniecznych przypadkach na jakość danych metodą neutralizacji zakłóceń danych bazy.

Model powstał po podzieleniu procesu neutralizacji na akcje, które mają charakter powtarzalny. Zróżnicowanie procesów wynika z odmiennych zasad i algorytmów realizacji poszczególnych akcji. Aby w ramach jednego modelu opisać ogół procesów przyporządkowano akcjom zestawy dopuszczalnych wariantów ich działania. Ze względu na małą objętość pracy scharakteryzowano jedynie te akcje, które są /mogą być/ realizowane automatycznie. Pozostałe grupy działań opisano ogólnie.

Każdy proces neutralizacji zakłóceń powinien spełnić trzy zadania: 1/ wykryć i zidentyfikować istniejące zakłócenie /a/, 2/ usunąć skutki jego wystąpienia oraz 3/ zlikwidować przyczyny jego powstania [9].

Poszczególnym zadaniom można przyporządkować odpowiadające im etapy prac neutralizacyjnych: kontrolę, korektę oraz konserwację.

Ponieważ etap konserwacji - ze względu na odmienny cel i charakter podejmowanych działań - nie wchodzi w skład procesu ochrony jakości bazy /realizowanego w trakcie eksploatacji użytkowej systemu/ pominięto jego opis w dalszych rozważaniach.

W ramach etapu k o n t r o l i wyróżniono cztery podstawowe akcje:

1/ przygotowanie kontroli,

2/ uruchomienie procedury testowania,

3/ testowanie danych,

4/ sygnalizacja zakłócenia /zakłóceń/.

Spośród wymienionych akcji jedna - przygotowanie kontroli - jest opcjonalna. Występuje jedynie wtedy, gdy kontrola ma charakter mieszany /manualno-programowy lub programowo-manualny/. Jej celem jest umożliwienie zrealizowania określonego algorytmu testowania. Spośród wariantów działania występujących w ramach tej akcji będzie nas interesować tylko jeden. Podczas manualnego testowania zawartości bazy danych porównuje się wartości bazy danych z wartościami danych na dokumentach. Przygotowanie kontroli polega na wyprowadzeniu wydruków kontrolnych zawartości bazy.

Podstawowa akcja kontroli - testowanie danych - może być uruchomiona w różny sposób. Wskazujemy kilka wariantów jej rozpoczęcia:

1/ z chwilą, gdy dana stała się przedmiotem działania określonych operacji przetwarzania,

2/ cyklicznie - w określonych wcześniej momentach czasowych,

3/ periodycznie - po dokonaniu na danych określonego typu ustalonej wcześniej liczby manipulacji mających charakter modyfikacji,

4/ w czasie, w którym nie są realizowane na bazie inne procesy użytkowe lub w czasie realizacji pewnych, ściśle określonych procesów,

5/ według innych reguł, np. przez próbkowanie [4][4].

Warianty pierwszy i ostatni zapewniają testowanie danych bazy wprowadzonych do niej za pomocą stosowanych komend Języka Manipulacji Danymi. W pozostałych wariantach testowanie dotyczy danych zawartych w bazie. Kontrole tej klasy określa się w literaturze mianem diagnostyki bazy danych [4,13]. Należy jednak podkreślić, że diagnozowanie bazy danych może się odbywać bez automatycznego wyzwalacza procedury testowania. Wówczas o czasie realizacji specjalnych programów diagnostycznych decyduje Administrator Bazy, zwykle na podstawie posiadanego harmonogramu /np. programy typu file balancing/ [6].

Najbardziej złożoną i zróżnicowaną akcją kontroli jest testowanie danych. Wskazanie wszystkich dopuszczalnych wariantów

[4]Obecnie warianty 2-5 służą przede wszystkim do celów klasycznej rewizji gospodarczej, rzadziej korzysta się z nich podczas kontroli poprawności danych bazy. Por. [6, s. 413-420].

stosowanych algorytmów należy uznać za niemożliwe. Testowanie danych jest możliwe dzięki temu, że są one elementami różnorodnych relacji. Testowanie to sprawdzanie, czy relacje te zachodzą. Związki te mogą być merytoryczne /pochodne w stosunku do relacji obiektów, które są opisywane przez dane/ lub formalne. W tym ostatnim przypadku wynikają one z charakterystyki danych dokonanej w trakcie projektowania struktury logicznej bazy. O ile zbiór relacji formalnych jest ograniczony i możliwy do bezpośredniej identyfikacji, o tyle zbiór relacji merytorycznych zależy od semantycznej zawartości bazy i może być bardzo rozległy. Chcąc zatem scharakteryzować akcję postanowiono skoncentrować się na jej trzech istotnych elementach. Odrębnie omówimy: przedmiot testowania, jego podstawę i klasy stosowanych algorytmów.

Przedmiotem badania może być: dana elementarna, dana złożona lub rekord bazy. Badanie może dotyczyć wszystkich danych zidentyfikowanych typów lub wybranych wystąpień danych określonych typów. Selektywne badanie powinno mieć - w przeciwieństwie do mechanizmu próbkowania - uzasadnienie merytoryczne [6, s.420].

Podstawą testowania jest zestaw danych, który wykorzystuje się w ramach określonego algorytmu do badania innych danych. W zestawie mogą znaleźć się: dane stałe, dane zmienne wchodzące w skład tego samego rekordu bazy bądź dane zmienne wchodzące w skład innych rekordów bazy.

Mianem algorytmu testowania określono funkcję, której realizacja umożliwia zerojedynkową identyfikację zakłóceń. Algorytmy stosowane do wykrywania zakłóceń merytorycznych można podzielić na trzy grupy:

1/ standardowe,
2/ standaryzowane,
3/ niestandardowe.

Uogólniający charakter klasyfikacji sprawia, że jest ona zdecydowanie odmienna od list wariantów działania wymienionych dla omawianych akcji. Wprowadza się ją, gdyż jest przydatna w analizie porównawczej. Kryterium podziału algorytmów stanowi poziom trudności ich opracowania.

Algorytmy standardowe to takie, które - w razie posługiwania się nimi - nie wymagają definiowania jakichkolwiek zmien-

nych parametrów. Znana jest funkcja badania, są ściśle określone: przedmiot testowania i jego podstawa[5].

Algorytmy standaryzowane wymagają wskazania podstaw badania, funkcja jest znana. Podstawa ta ściśle zależy od przedmiotu kontroli[6].

Algorytmy niestandardowe to takie, które dla określonego przedmiotu testowania wymagają określenia zarówno funkcji porównania, jak i jej podstawy. Dotyczą one przede wszystkim relacji merytorycznych, w których uczestniczą określone dane bazy.

Ostatnią akcją kontroli jest sygnalizowanie zakłóceń. Odbiorcą informacji o typie zniekształceń oraz o danych, w których wystąpił błąd, jest osoba dokonująca korekty[7]. W przeciwieństwie do poprzednich akcji zbiór dopuszczalnych wariantów sygnalizacji można wyznaczyć dzięki kryteriom klasyfikacji podanym w literaturze przedmiotu [2]. Jej dopuszczalne warianty zidentyfikowano po wskazaniu zbiorów dopuszczalnych wartości dla poszczególnych istotnych kryteriów:

1/ ze względu na technikę - programowa /ograniczenie zgodne z przyjętym wcześniej/,

2/ ze względu na sposób stosowania - wyłącznie obligatoryjna,

3/ ze względu na charakter - jedynie uprzedzająca,

4/ ze względu na formę - graficzna trwała lub nietrwała,

5/ ze względu na tryb - natychmiastowa /praca on-line/ lub opóźniona /aktualizacja wsadowa/,

---

[5]Dla tej grupy algorytmów reprezentatywne są następujące metody badania: testowanie cyfry kontrolnej, poziomej sumy kontrolnej, klasy pola, występowania duplikatów wartości pola /pól/, występowania zer lub spacji.

[6]Typowymi przedstawicielami tej grupy algorytmów są: testowanie według listy wartości lub według zanegowanej listy, badanie przedziałów wartości lub ich negacji, testowanie rekordów zmiennej długości według wbudowanych liczników i/lub ograniczników.

[7]W pracy [4] stwierdzono celowość trwałego rejestrowania zakłóceń występujących w systemie. Umożliwiałoby to prowadzenie statystycznej analizy błędów według typów i źródeł powstania. Mimo że autor podziela ten pogląd, postanowiono nie poddawać analizie porównawczej tego wariantu akcji. Ponieważ w innych publikacjach nie potwierdzono występowania tej opcji w ramach procesów neutralizacji, wariant ten uznano za propozycję teoretyczną.

6/ ze względu na zasięg - charakterystyka błędu lub sytuacji błędowej[8].

Zróżnicowanie trzech ostatnich cech akcji nie decyduje o różnicowaniu stosownych programowych procedur sygnalizacji.

Kolejny etap procesu neutralizacji - k o r e k t ę - również można przedstawić jako ciąg akcji realizowanych w określonej sekwencji. Zdecydowano się wyróżnić cztery podstawowe kroki proceduralne korekty:

1/ pierwotna reakcja na wystąpienie zakłócenia,

2/ sterowanie korekty,

3/ usunięcie pierwotnych błędów poza systemem bazy danych,

4/ usunięcie skutków zakłóceń z bazy danych.

Spośród nich jedynie dwa - pierwszy i czwarty - będą przedmiotem dalszej analizy. Sterowanie korekty to zbiór działań o charakterze decyzyjnym. Decyzje takie mogą dotyczyć zasad realizacji zarówno trzeciej, jak i czwartej akcji. Ponieważ - zgodnie z opisem zawartym w literaturze przedmiotu - ogół tych działań ma charakter manualny, zrezygnowano ze szczegółowego ich badania [2,9].

Krok trzeci ma odmienny charakter. Zdecydowana większość zakłóceń merytorycznych jest skutkiem błędów popełnianych w trakcie wypełniania dokumentów, ich przenoszenia na MNI /w przypadku wsadowej aktualizacji bazy/ lub ich wprowadzania do bazy /w przypadku aktualizacji on-line/. W związku z tym za klasyczną należy uznać sytuację, w której w celu ustalenia właściwych wartości danych lub poprawy innych błędów jesteśmy zmuszeni powrócić do etapów poprzedzających przetwarzanie i - jeżeli do usunięcia zakłóceń z bazy chcemy wykorzystać programy zakładania i/lub aktualizacji - powtórzyć proces przetwarzania na danych poprawionych od miejsca, w którym popełniono pierwotny błąd. Ponieważ środki programowe, które stosuje się w takich przypadkach, nie operują na bazie danych, szczegółowy opis dopuszczalnych wariantów realizacji tego kroku uznano za zbędny.

Pierwotna reakcja na zakłócenia może być zróżnicowana. Jeżeli zakłócenie wykryto w rekordzie bazy do niej wprowadzonym, to można wskazać następujące warianty działania:

[8] Kryteria tej klasyfikacji i szczegółowy opis poszczególnych wariantów ich wartości można znaleźć w [2, s. 68-76].

1/ brak reakcji /oczywiście poza sygnalizacją/,

2/ odrzucenie rekordu zawierającego błędne dane,

3/ oznaczenie danych błędnych,

4/ zablokowanie zniekształconych danych bazy.

W wariancie drugim skutek użytkowy działania może być różny. Jeżeli rekord jest dopisywany do bazy, to taka reakcja może powodować, że będzie się on znajdować poza bazą do czasu powtórnego wprowadzenia po dokonaniu poprawek. Jeżeli rekord jest modyfikowany, to w bazie pozostaje jego poprzednia wersja.

W pozostałych przypadkach rekord zawierający błędy "przedostaje się" do bazy, a podejmowane działania zmierzają do zminimalizowania negatywnego wpływu zakłóceń na wyniki procesów użytkowych realizowanych do momentu dokonania pełnej korekty [4].

Jeżeli zidentyfikowano zakłócenie w ramach rekordu znajdującego się w bazie danych, to dopuszczalne są pierwsza, trzecia i czwarta spośród wymienionych reakcji [4].

Usuwanie skutków zakłóceń merytorycznych z bazy danych można realizować w różny sposób. O wyborze właściwego wariantu decyduje zbiór wcześniejszych działań podjętych w ramach neutralizacji i charakterystyka zasad aktualizacji określonej danej. Jeżeli rekord zawierający błędy został uprzednio odrzucony, to w celu ostatecznej korekty można spożytkować własne programy zakładania i/lub aktualizacji. Jeżeli rekord ulokowany w bazie kojest korygowany, a aktualizacja polega na podstawieniu /zamianie wartości na inną/, to również poprawki można wprowadzić za pomocą tych programów. Pozostałe przypadki są bardziej złożone, szczególnie korekta zakłóceń danych wykrytych w trakcie diagnozowania bazy. Ich algorytmiczna unikalność wymaga zastosowania zindywidualizowanych procedur korekty, ukierunkowanych na konkretne sytuacje błędowe.

Omawiany model procesu neutralizacji, który jest uogólnieniem zbioru faktycznie realizowanych procesów neutralizacji, będzie podstawą analizy porównawczej i diagnozy stanowiącej cel artykułu.

3. Diagnoza zasobów systemu zarządzania bazą wykorzystywanych do neutralizacji zakłóceń merytorycznych

Szczegółowym przedmiotem badań jest USZBD RODAN wersja 3.0. W artykule traktuje się go jako typowy i klasyczny przykład zastosowania modelu systemu opracowanego przez DBTG CODASYL do

sieciowych baz danych [3]. Niezbędne informacje na temat możliwości tego systemu zaczerpnięto z dokumentacji dostarczonej przez producenta [10].

Mianem zasobów systemu określono zbiór środków programowych bezpośrednio dostępnych w jego głównych elementach. Oto one: JOD, JMD, Procesor Funkcji Pomocniczych, Wielozadaniowy Monitor Bazy Danych i zbiór niezależnych programów pomocniczych.

W wyniku porównania wcześniej ustalonego zbioru wariantów akcji z tymi zasobami ustalono, że parametryczną realizację wybranych wariantów akcji umożliwiają: JOD /Język Opisu Danych/, JMD /Język Manipulacji Danymi/ i Procesor Funkcji Pomocniczych.

W obrębie JOD będą to:

1/ klauzula CHECK podopisu danej w ramach opisu rekordu,

2/ klauzula DUPLICATES podopisu rekordu, rekordu nadrzędnego lub podrzędnego,

3/ klauzula NULL dotycząca kluczy sortowania w ramach podopisu rekordu podrzędnego.

Pewne dodatkowe możliwości stwarza klauzula ON [ERROR DURING] wchodząca w skład podopisu rekordu lub podopisu danej. Umożliwia ona bowiem bardziej szczegółową i wszechstronną sygnalizację zakłócenia niż mechanizm powszechnie stosowany w ramach komend JMD /klauzula ON ERROR GO TO..../. Różnica między nimi polega na tym, że procedura bazy danych wywoływana w pierwszym przypadku na dostęp do dodatkowych informacji identyfikujących błędne dane [10].

Klauzula CHECK umożliwia testowanie danych podczas wykonywania komend aktualizacji danych merytorycznych /STORE, MODIFY/. Za algorytmy badania nadające się do parametrycznego wykorzystania można uważać: badanie klasy znaków pola /opcja PICTURE/ i testowanie według list wartości lub ich przedziałów /mogą one być zanegowane - VALUE NOT/[9]. W ramach tejże klauzuli istnieje możliwość dołączania procedur bazy danych, które mogą opisywać inne algorytmy. Istnieje jednak ważne ograniczenie ich

[9] W podstawowym tekście artykułu pominięto ograniczenie występujące w przypadku opcji VALUE - liczba wartości lub ich przedziałów nie może przekraczać 10. Uznano je za cechę dotyczącą wyłącznie systemu RODAN niezbyt istotną na tym poziomie szczegółowości rozważań.

grupy. Otóż, podstawą testowania mogą być jedynie dane stałe lub zmienne wchodzące w skład kontrolowanego rekordu.

Włączenie do stosownego podopisu klauzuli DUPLICATES umożliwia realizację standardowego algorytmu badania powtarzalności wybranych pól lub ich grup. Chodzi wyłącznie o pola służące do porządkowania rekordów członkowskich w ramach setu /użycie klauzuli w ramach klauzul ORDER lub KEY/, o pola będące kluczami wyszukiwania /wykorzystanie w ramach klauzuli SEARCH/ lub kluczami CALC /klauzula LOCATION MODE/.

Wymienione klauzule działają w ramach jednorodnego mechanizmu neutralizacji. Jest nim automatyczna realizacja określonych wariantów akcji kontroli i wariantu wybranego spośród wariantów pierwszej akcji korekty w trakcie wykonywania komend JMD aktualizujących zawartość bazy danych. Oto jego charakterystyka:

1/ testowanie dotyczy każdego wystąpienia danych określonego typu w momencie, gdy znajduje się ono na wejściu do bazy,

2/ realizuje się algorytm testowania zadeklarowany w schemacie bazy, równocześnie rejestruje się wszystkie elementarne operacje wpływające na stan zawartości bazy danych w tzw. Tablicy Odwracania Komend /TOK/,

3/ jeżeli testowanie zakończyło się wynikiem negatywnym /wykryto zakłócenie/, to wszystkie zmiany zawartości bazy dokonane w ramach komendy są likwidowane na podstawie TOK - tym samym wprowadzany lub zaktualizowany rekord odrzuca się, gdyż uznaje się go za błędny,

4/ do specjalnego pola systemowego włączanego do wszystkich programów użytkowych realizowanych na bazie /RPDMS/ przesyła się informacje identyfikujące zakłócenie,

5/ istnienie opcji ON ERROR lub klauzuli ON ERROR powoduje automatyczne przejście do procedury sygnalizacji lub jej wywołanie.

Ostatnim spośród zidentyfikowanych zasobów kontrolnych jest funkcja pomocnicza PRINT RECORD wchodząca w skład Procesora Funkcji Pomocniczych. Wykorzystanie jej w postaci PRINT RECORD...DATA umożliwia wyprowadzenie określonej części zawartości bazy danych w formie pozwalającej na manualną kontrolę [10]. Można wobec tego uznać, że system zarządzania spełnia wymagania stawiane przez pierwszą akcję kontroli.

Po porównaniu powyższych zasobów z przedstawionymi wcześniej wymaganiami można wskazać następujące rozbieżności:

1/ system rozpoczyna kontrolę tylko po pojawieniu się danej określonego typu na wejściu do bazy,

2/ do mechanizmu neutralizacji wbudowanego w system nie można włączyć algorytmów testowania, których podstawą są zmienne dane bazy spoza badanego rekordu,

3/ nie stosuje się standardowych komunikatów do celów sygnalizacji zakłóceń, wymienione wcześniej zasoby jedynie wspomagają sygnalizację,

4/ w ramach mechanizmu neutralizacji obligatoryjny charakter ma odrzucenie błędnej aktualizacji jako pierwotna reakcja na zakłócenie.

Można zatem sądzić, że w systemie istnieje wiele znaczących luk. Wydaje się jednak, że taka ilościowa analiza nie wystarcza do postawienia dokładnej diagnozy. Dotychczas bowiem nie uwzględniono:

1/ częstości stosowania poszczególnych wariantów akcji,

2/ możliwości wykorzystania własnych środków programowych,

3/ charakteru funkcji ochrony jakości bazy.

Należy zwrócić uwagę, że obowiązkowe w ramach istniejącego mechanizmu neutralizacji warianty akcji uruchamiania kontroli i pierwotnej reakcji na zakłócenia są wariantami klasycznymi, wykorzystywanymi dotychczas w decydującej większości przypadków. Stosowanie kontroli nie na "wejściu do bazy" i reakcji innych niż "odrzucenie błędnego rekordu" wymaga zaniechania schematycznego myślenia projektantów systemów ochrony jakości bazy. W żadnym spośród nazwanych i opisanych w dostępnej literaturze przedmiotu systemów zarządzania nie są one wbudowane w standardowy mechanizm neutralizacji.

Opisane wykorzystywane parametrycznie algorytmy testowania należą do grupy najczęściej stosowanych. Prawie tak samo często korzysta się jedynie z wybranych spośród pozostałych standardowych i standaryzowanych algorytmów. Algorytmy, w których za podstawę porównania przyjmuje się dane zmienne pochodzące z innych rekordów bazy, są wyłącznie algorytmami niestandardowymi, bardzo rzadko stosowanymi w rzeczywistych procesach neutralizacji z powodu swojej złożoności i czasochłonności.

Wypełnianie funkcji zarządzających wobec bazy danych jest możliwe dzięki:

1/ standardowym procedurom istniejącym w ramach systemu zarządzania bazą danych lub

2/ procedurom własnym zbudowanym na podstawie języków bazowych i/lub JMD.

Jest oczywiste, że wykorzystanie do celów ochrony jakości bazy parametrycznych procedur i standardowego mechanizmu neutralizacji systemu zarządzania bazą danych jest rozwiązaniem, które ma wiele zalet. Istotne są zwłaszcza korzyści wynikające z możliwości zastosowania równoważnego funkcjonalnie programu /podprogramu/ standardowego zamiast procedur własnych.

Z punktu widzenia Administratora szczególnie istotne znaczenie ma wbudowanie standardowego mechanizmu neutralizacji w system zarządzania bazą. Zastąpienie go procedurami własnymi mogłoby być zadaniem przekraczającym możliwości projektanta systemu ochrony. Chodzi głównie o procedury odwracania skutków realizacji komend JMD i zbierania /grupowania/ zestawu istotnych informacji o zakłóceniu.

Analiza wariantów realizacji akcji nie mających odpowiedników w postaci procedur systemu zarządzania wykazała, że ich zdecydowana większość może być realizowana jako procedury własne ulokowane w programach użytkowych lub procedury bazy danych deklarowane w schemacie.

Należy stwierdzić, że system zarządzania bazą danych spełnia minimalne wymagania w zakresie ochrony jakości bazy. Minimum to stanowi specyficzny dla bazy danych uniwersalny mechanizm neutralizacji zakłóceń jakości bazy. Istnieją merytoryczne przesłanki rozbudowy i uelastycznienia tego mechanizmu przez wbudowanie doń stosownych opcji /szczególnie odmiennych wariantów pierwotnej reakcji na zakłócenia/. Ocena pozostałych zidentyfikowanych zasobów systemu zarządzania bazą nie może być jednoznaczna. Można uznać, iż udowodniono, że opisany zbiór procedur nie jest zbiorem kompleksowym; wykorzystywanie wyłącznie procedur standardowych nie zawsze umożliwia pełną realizację procesów neutralizacji. Wbudowanie do systemów zarządzania dodatkowych opcji nie powinno stwarzać większych trudności /szczególnie jeżeli chodzi o sparametryzowanie zbioru standardowych i standaryzowanych algorytmów testowania/.

Jednak uzupełnienie takie nie jest konieczne ze względu na alternatywne możliwości opracowania i wbudowania odpowiednich procedur do systemu ochrony jakości bazy.

Rozbudowa zdolności funkcjonalnych systemu zarządzania byłaby rozwiązaniem ułatwiającym pracę Administratorowi Bazy - twórcy systemu ochrony. Argument tego typu nie zawsze ma decydujące znaczenie. Za podstawowy kontrargument uznaje się, że funkcja ochrony jakości bazy nie może być zaliczana - niezależnie od jej istotności - do podstawowych zadań systemu zarządzania, tj. takich, które warunkują eksploatację bazy. Postulaty rozbudowy mogą dotyczyć także innych funkcji tego systemu /przynajmniej równie istotnych/. W związku z tym decyzja o wbudowaniu weń dodatkowych procedur kontroli może być podjęta jedynie na podstawie kompleksowej analizy postulowanych zmian funkcjonalnych systemu jako całości, a nie tylko na podstawie diagnozy jednej szczegółowej funkcji realizowanej w jego ramach.

LITERATURA

[1] R.J. B u h r, B. P a g u r e k, D.A. T h o m a s, Validation Algorithms for Pointer Values in DBTG Databases, [w:] "ACM Transactions on Database Systems", 1977, nr 12, s. 352-369.

[2] B. B u ś k o, H. F i l i p e k, J. S l i w i e ń s k i, Wiarygodność informacji ekonomicznej w systemach informatycznych, Warszawa 1980, PWE.

[3] CODASYL, Język Opisu Danych, red. W. Staniszkis, Warszawa 1977, OBRI.

[4] G.C. E v e r e s t, Database Management. Objectives, Organisation and System Functions, Minneapolis 1974 /maszynopis powielany/.

[5] K. K u b ó w, M. S k w a r n i k, Analiza możliwości racjonalizacji procesu projektowania systemu zabezpieczeń jakości danych - wybrane zagadnienia metodologiczne, [w:] Informatyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu /w druku/.

[6] J. M a r t i n, Security, Accuracy and Privacy in Computer Systems, Englewoods Cliffs 1973.

[7] W. S i n a w i n a, Ocenka kačestva funkcjonirovanija ASV, Moskva 1973, Ekonomika.

[8] M. S k w a r n i k, Proces odtwarzania bazy danych - przegląd wybranych metod, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 183, Wrocław 1981.

[9] A. S t r a m o w s k i, Odtwarzanie stanu bazy danych - metody i techniki, w: SPIS - 77. Materiały konferencyjne, Warszawa 1977, GUS.

[10] USZBD RODAN Wersja 3.0, dokumentacja systemu, t. 1-4, Warszawa 1980, CPiZI.

ANALYSIS OF PROSPECTS FOR UTILIZING RESOURCES OF A DATA BASE MANAGERIAL SYSTEM IN THE PROCESS OF BASE QUALITY PROTECTION

Summary

Resources of a data base managerial system, enabling the neutralization of base substantial disturbances, are analyzed in the paper. The previously developed general model for neutralizing such errors is also described. Base management system resources, which may be used to supervise and correct the contents of the base, are determined here and a comparative analysis is carried out. The main conclusion drawn here is that the base management system fulfills only minimum application requirements concerning base quality protection.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Mirosław D y c z k o w s k i
Akademia Ekonomiczna we Wrocławiu

ANALIZA PRZYDATNOSCI SRODKÓW OCHRONY DOSTĘPU DO DANYCH USZBD RODAN DLA SYSTEMU ZARZĄDZANIA WSPOMAGANEGO KOMPUTEREM

1. Wprowadzenie

W ostatnich latach nastąpił w świecie dynamiczny rozwój systemów informatycznych wspomagających procesy zarządzania w organizacjach gospodarczych. Było to możliwe dzięki dostatecznemu opanowaniu wielowariantowych rozwiązań zarówno technologii bazy danych, jak i związanego z nią teleprzetwarzania[1]. W konsekwencji wzrosło zainteresowanie problemami szeroko rozumianej ochrony danych, która zyskała nowe znaczenie w warunkach systemów bazy danych, obsługujących równocześnie wielu użytkowników zdalnych i lokalnych o różnych zakresach uprawnień do dostępu i manipulowania "wspólnymi" zasobami danych.

Celem niniejszego artykułu jest określenie stopnia przydatności środków ochrony dostępu do danych zawartych w rozpowszechnianym w kraju Uniwersalnym Systemie Zarządzania Bazą Danych RODAN dla systemów zarządzania wspomaganego komputerem

[1] Rozwój technologii bazy danych i teleprzetwarzania umożliwił dostarczenie odpowiednich narzędzi do realizacji lansowanej pod hasłami MIS, IMIS, corporate planning itp. idei kompleksowych systemów zarządzania wspomaganego komputerem. Pozwolił także na wyłonienie się bardziej rozwiniętych wersji tych systemów, jak systemy typu ekspert czy komputerowego wspomagania procesów decyzyjnych /decision support systems/.

/SZWK/[2]. Za przedmiot analizy przyjęto system RODAN, gdyż w pracy omawia się wyłącznie systemy bazy danych[3], rozważa się także jeden aspekt szeroko rozumianej ochrony danych, a mianowicie ochronę dostępu do danych. Mała objętość artykułu zmusza autora do marginalnego traktowania łączących się z nią problemów, takich jak: integralność danych i bazy danych, poufność danych, odtwarzanie stanów bazy danych.

2. System zarządzania wspomaganego komputerem jako przedmiot ochrony dostępu do danych

Rozważania poprzedzimy sprecyzowaniem podstawowych wymagań, jakie stawia się ochronie danych w systemach zarządzania wspomaganego komputerem realizowanych w technologii bazy danych. W tym celu należy zdefiniować funkcje systemów tej klasy. Wydaje się, że na potrzeby artykułu wystarczy następujący wykaz głównych funkcji SZWK:

- utrzymywanie i aktualizowanie bazy danych systemu i zbioru modeli typowych i indywidualnych procesów decyzyjnych,
- przetwarzanie bazy danych z zastosowaniem procedur opartych na metodach matematyczno-ekonomicznych /symulacji, modelowania, wariantowania, optymalizacji itp./ i wykorzystaniem modeli procesów decyzyjnych,
- dostarczanie decydentom - kierownictwu obiektu wielowariantowych wyników umożliwiających podejmowanie właściwych decyzji indywidualnych i złożonych w czasie dostosowanym do danego poziomu zarządzania,
- generowanie decyzji zrutynizowanych /typowych i prostych/ i w ten sposób osłabienie oddziaływania zasady ograniczonej rozpiętości kierowania.

---

[2] Artykuł jest kolejną pracą autora na temat ochrony danych w systemach zarządzania wspomaganego komputerem, realizowanych w technologii bazy danych. Wykorzystano w nim m. in. wyniki prowadzonych w Akademii Ekonomicznej we Wrocławiu prac badawczych /w których autor brał udział/ w ramach problemu węzłowego 06.4, które dotyczyły wspomagania zarządzania transportem kolejowym w Hucie Katowice, a także wcześniejszych prac nad eksperymentalną bazą danych KADRY. Patrz [2-5;8].

[3] Wiąże się to z przekonaniem autora, że większość SWKZ będzie realizowana w technologii bazy danych opartej na komercyjnych SZBD.

W omawianej klasie systemów istnieją szczególne wymagania względem ochrony danych. Bez identyfikacji potencjalnych zagrożeń ochrony[4] można określić główne przyczyny opisanej sytuacji. Są nimi:

- wzrastające uzależnienie organizacji gospodarczych od sprawnego funkcjonowania systemów wspomagających zarządzanie, a zwłaszcza jakości[5] ich zasobów informacyjnych,
- wartość rzeczywista i cenność użytkowa danych przechowywanych i przetwarzanych w systemie i operujących na nich procedur,
- uwzględnienie potrzeb użytkownika nie będącego informatykiem, ułatwienie mu dostępu i korzystania z zasobów systemu,
- wielodostępność i funkcjonowanie jako system o działaniu bezpośrednim oraz różny zakres uprawnień do korzystania z dostępnych trybów pracy systemu przez różne grupy użytkowników[6],
- dodatkowe wymagania wynikające z przepisów o ochronie tajemnicy państwowej i służbowej oraz dotyczące zabezpieczenia indywidualnych danych personalnych przed nieupoważnionym dostępem i rozpowszechnianiem.

Systemy zarządzania wspomaganego komputerem ze względu na swe funkcje pełnione w obiektach gospodarczych powinny działać ciągle, sprawnie i nadążnie. Jest to możliwe tylko wówczas, gdy nie występują zewnętrzne zakłócenia pracy systemu, w tym brak sterowania dostępem do jego zasobów. Nieupoważniony dostęp i manipulowanie zawartością bazy danych lub proceduramI SZWK mogą stać się przyczyną naruszenia jakości informacji /danych/ systemu i obniżenia walorów użytkowych. Tak więc wbudowanie do SZWK systemu ochrony dostępu do danych jest warunkiem koniecznym, choć oczywiście nie gwarantującym realizacji jego celów.

---

[4] Szczegółową identyfikację, analizę i ocenę zagrożeń ochrony dostępu do danych w SZWK autor przedstawił w [3].

[5] Pod pojęciem jakości informacji /danych/ autor rozumie ich poprawność, aktualność, kompletność i niesprzeczność /spójność/.

[6] Często formułuje się tezę, że systemy zarządzania wspomaganego komputerem nie muszą być systemami o działaniu bezpośrednim, gdyż wystarczy dostarczenie przez nie informacji wynikowych we "właściwym" czasie. Doświadczenie pokazuje jednak, że "właściwy" czas znaczy prawie zawsze tyle co na bieżąco.

Baza danych systemu i zbiory procedur wyszukujących, przetwarzających i edycyjnych zawierają informacje służące do analizy i modelowania podstawowych procesów techniczno-ekonomicznych sterowanego obiektu gospodarczego. Informacje te są adresowane do decydentów należących najczęściej do tzw. naczelnego kierownictwa obiektu i opisują w formie syntetycznej, zagregowanej i przekrojowej aktualną sytuację finansową, produkcyjną, kadrową, rynkową itp. przedsiębiorstwa i warianty jej przewidywanego rozwoju. Mają one zatem wysoką cenność użytkową, a także wartość rzeczywistą związaną z nakładami pracy przeznaczonymi na ich zebranie, przechowywanie, przetwarzanie. W związku z tym nie powinny ulec zniszczeniu, uszkodzeniu lub dostać się w niepowołane ręce.

Podczas opracowywania SZWK dokłada się wszelkich starań, aby był on łatwy w eksploatacji i dostępny dla wszystkich, którzy mają korzystać z jego pomocy, gdy rozwiązują problemy decyzyjne. Można mówić o prostocie dostępu i realizacji zwłaszcza zapytań i transakcji, gdyż użytkownikiem systemu jest personel kierowniczy obiektu, który nie musi znać języków programowania[7] i zasad operowania zadaniami.

System zarządzania wspomaganego komputerem projektuje się najczęściej jako system wielodostępny, o działaniu bezpośrednim. Z jego zasobów może jednocześnie korzystać wielu użytkowników o odmiennych uprawnieniach do dostępu i manipulowania określonymi typami struktur danych i procedur. W tak zorganizowanym systemie środki ochrony dostępu do danych powinny zapewnić, by żaden użytkownik nie mógł:

- dotrzeć do zasobów systemu, a w szczególności do danych lub programów innych użytkowników bez właściwych upoważnień,
- manipulować zasobami systemu, a zwłaszcza danymi i programami innych użytkowników bez stosownych zezwoleń,
- bez właściwego upoważnienia pozbawić innych użytkowników możliwości korzystania z zasobów informacyjnych i procedur systemu zgodnie z posiadanymi przezeń uprawnieniami.

Podane założenia, dodatkowe wymagania wynikające z przepisów o ochronie tajemnicy państwowej i służbowej oraz zasady zabez-

[7] Wystarczy znać proste języki zapytań typu SEQUEL i zasady operowania transakcjami przy korzystaniu z procesora komunikatów.

pieczenia indywidualnych danych personalnych stanowią grupę czynników determinujących ochronę informacji w SZWK.

Na podstawie analizy syntetycznego opisu można stwierdzić, że w systemach tej klasy zabezpieczenia wymagają wszystkie ich elementy składowe /zwłaszcza baza danych/, wszystkie realizowane funkcje użytkowe i technologiczne /zwłaszcza wyszukiwanie, aktualizowanie i edycja zestawień wynikowych/, a także że zakres zadań, struktura i funkcjonowanie systemu ochrony danych wymagają różnorodnych środków ochrony zapewniających m. in. autoryzację dostępu i sterowanie manipulowaniem elementami danych.

Oczywiście dobór środków ochrony określonych elementów SZWK i rozwiązania organizacyjne, techniczne i technologiczne samego systemu ochrony dostępu do danych wymagają przeanalizowania i uwzględnienia m. in. takich czynników, jak:

- model i typ zarządzania obiektem,
- struktura organizacyjna,
- struktura zarządzania i powiązania informacyjno-decyzyjne,
- warianty zarządzania zasobami informacji,
- poziom SZWK,
- cenność poszczególnych elementów danych,
- koszt realizacji systemu,
- czas przygotowania i realizacji.

Bez szczegółowej analizy wymienionych czynników można przyjąć, że w systemach zarządzania wspomaganego komputerem powinien funkcjonować system ochrony dostępu do elementów danych i procedur oraz ochrony wykorzystania danych przez operujące zadania użytkowe. System ten należy tak skonstruować, aby posiadającym uprawnienia zapewnił dostęp i możliwość manipulowania danymi i procedurami. Powinna go zatem cechować dostateczna selektywność[8] oraz wnikliwość[9]. Działanie jego nie może obciążać

[8] Selektywność ochrony jest parametrem, za którego pomocą ocenia się związki typu system ochrony-chronione elementy danych. Ponadto służy do określenia, na jakich poziomach i w jakich zakresach funkcjonują poszczególne mechanizmy ochrony dostępu, a tym samym determinuje możliwości ich stosowania do określonych typów elementów struktury danych. Por.[4;9].

[9] Wnikliwość ochrony, podobnie jak selektywność, zapewnia ocenę związków typu system ochrony-chronione elementy danych. Służy ona do określenia typów operacji, które mogą być objęte przez określony mechanizm lub cały system zabezpieczeń wspólnym kryterium dopuszczalności wykonania, wyznaczonym dla danego poziomu i zakresu ochrony. Por.[3;9].

zbytnio czasu realizacji zadań użytkowych. Wystarczająca powinna być jego przepustowość /liczona najczęściej dla średniej długości kolejki użytkowników i przeciętnego jednostkowego czasu obsługi/. Nie może ona jednak obniżać niezawodności i skuteczności działania systemu ochrony poniżej żądanego minimum. Łatwe powinno być także włączenie go do zespołu zabezpieczeń danych i procedur zarówno programowych, jak i sprzętowych, technicznych oraz organizacyjnych.

Na podstawie dotychczasowych rozważań spróbujemy określić podstawowe wymogi w zakresie ochrony dostępu do danych w systemach zarządzania wspomaganego komputerem [por. 7, s. 272-273]:

1. Przechowywane i przetwarzane w systemie dane i procedury powinny być:

- chronione, przy czym zabezpieczenia powinny funkcjonować na wszystkich poziomach struktur danych w zakresie maksimum typu elementu struktury i zapewniać dostateczną selektywność i wnikliwość ochrony,
- sprawdzone pod względem manipulowania nimi - chodzi zwłaszcza o możliwość uzyskania dostępu w sposób inny niż standardowe ścieżki selekcji i przetwarzanie danych jawnych,
- kontrolowane - konieczne jest prowadzenie stałej działalności audytorskiej,
- odtwarzalne - zagadnienie to nie jest wprawdzie elementem ochrony dostępu do danych, lecz jego związki z naruszeniami zabezpieczeń dostępu i manipulowania danymi są oczywiste.

2. Użytkownicy systemu powinni być:

- identyfikowani - ale identyfikacja powinna być wielopoziomowa, charakteryzująca się niską wartością współczynników błędnej akceptacji i błędnego odrzucenia[10].

3. Działania użytkowników powinny być:

- autoryzowane - chodzi o zgodność działania użytkowników z przyznanymi im uprawnieniami,
- monitorowane - jest to niezbędny warunek identyfikacji użytkowników naruszających ograniczenia systemu ochrony i sposobów ich działania, co w połączeniu z prowadzeniem statystyk ochrony pozwala na konserwację i modernizację systemu zabezpieczeń.

[10] W pracy [5] przedstawiono koncepcje identyfikacji użytkowników w systemie nauczania wspomaganego komputerem OSKAR. Można ją zastosować również w SZWK.

4. System ochrony dostępu do danych powinien się cechować:
- niezawodnością,
- przepustowością,
- skutecznością,
- efektywnością i ekonomicznością ochrony[11].

Opisane wymagania będą stanowić podstawę oceny przydatności środków ochrony dostępu do danych dostarczonych przez USZBD RODAN dla systemów omawianej klasy.

3. Środki ochrony dostępu do danych USZBD RODAN - przegląd i analiza

Uniwersalny System Zarządzania Bazą Danych RODAN należy do grupy systemów sieciowych opartych na specyfikacjach i zaleceniach komitetów CODASYL[12]. Także zastosowane w nim środki ochrony dostępu do danych są zgodne z raportami DBTG z kwietnia 1971 i DDLC z czerwca 1973 roku i przez to zbliżone do rozwiązań przyjętych w innych codąsylowskich SZBD. Stwierdzenie to pozwala jeszcze przed szczegółową analizą mechanizmów systemu RODAN postawić tezę roboczą, że znaczne rozpowszechnienie systemów codasylowskich - zwłaszcza takich jak IDMS, DMS-1100, DMS-90, DBMS-10, DBMS-20 - także w aplikacjach z zakresu zarządzania wspomaganego komputerem świadczy o przydatności zastosowanych w nich zabezpieczeń dla tej klasy systemów[13]. W celu jej udowodnienia zidentyfikujmy i oceńmy możliwości USZBD RODAN w dziedzinie ochrony dostępu do danych i ich zgodności ze sformułowanymi w poprzednim punkcie wymaganiami.

Powszechnie uznano, że w praktyce ochrona dostępu do danych por.[2;7;8] polega na włączeniu do systemu zarządzania bazą danych i środków z nim bezpośrednio współdziałających oraz go wspo-

---

[11] Powyższe parametry systemu ochrony dostępu do danych zależą od rozwiązań implementacyjnych przyjętych w konkretnym SZWK. Z tego względu analiza ich dla USZBD RODAN będzie prowadzona w ograniczonym zakresie.

[12] Podstawą dalszych wywodów w części dotyczącej opisu mechanizmów systemu RODAN jest [10].

[13] Wprawdzie w większości przypadków wyższe są parametry jakości i niezawodności otoczenia systemu ochrony dostępu do danych, zwłaszcza systemu operacyjnego, sprzętu i przyjętych rozwiązań organizacyjnych, lecz nie zmienia to prawdziwości postawionej tezy w odniesieniu do SZBD.

magających takich mechanizmów programowych lub przyjęciu takich rozwiązań organizacyjnych[14], które by zapewniały realizację trzech podstawowych grup funkcji:

1. identyfikacji i autoryzacji użytkownika /rozumianego często w sensie zadania użytkowego/ i związanych z nim warunkowych i bezwarunkowych praw dostępu,

2. sterowania dostępem zgodnie ze zbiorami ograniczeń dostępu określonymi dla poszczególnych użytkowników i elementów danych, do których żądają dostępu,

3. ograniczenia możliwości wykorzystania danych w przypadku przełamania lub obejścia systemu ochrony przez stosowanie różnych metod enkrypcji danych /w tym konwersji i kompresji danych jako metod quasi-enkrypcji/ por.[2;9].

Realizacja pierwszej grupy funkcji w przypadku pracy wsadowej jest zapewniona przez wbudowane w Język Opisu Danych /JOD/ schematu i podschematu mechanizmy blokad dostępu /klauzule PRIVACY LOCK/ i współdziałające z nimi, wchodzące w skład Języka Manipulacji Danymi /JMD/ klucze dostępu /komenda PRIVACY KEY/. Mechanizmy blokad i kluczy dostępu są tylko w pewnym stopniu zorientowane na identyfikację i autoryzację zadań użytkowych, a ich podstawowym zadaniem jest zapewnienie sterowania dostępem na poziomie schematu i podschematu oraz Wielozadaniowego Monitora Bazy Danych /WMBD/, dlatego też zostaną one omówione przy analizie drugiej grupy funkcji.

Funkcje identyfikacji i autoryzacji użytkownika są znacznie szerzej spełniane w pracy bezpośredniej, także w środowisku telekomunikacyjnym, w którym odpowiednie środki zabezpieczające są wbudowane w Monitor Teleprzetwarzania /MTP/ systemu RODAN. MTP umożliwia opcjonalnie:

- identyfikację operatora /grupy operatorów/ terminala za pomocą przydzielonego mu kodu,

- autoryzację dostępu operatora /grupy operatorów/ terminala dzięki hasłu kontrolnemu,

- określenie poziomu dostępu operatora /grupy operatorów/ terminala, a więc umożliwienie wykonywania tylko określonych transakcji,

[14] Możliwości organizacyjne wiążą się z określeniem przez zespół administratora bazy danych przy współudziale administratora zastosowań różnych wariantów strategii ochrony danych.

- określenie poziomu ochrony transakcji w celu umożliwienia sterowania dostępem i wykonywaniem transakcji wymagających ochrony.

Trzy pierwsze opcje są definiowane makroinstrukcją MTPURT generującą tablicę kontroli operatorów. Oto postać makroinstrukcji:

```
MTPURT  TYPE={INITIAL,ENTRY,FINAL} [,SUFFIX=xx]
        [,USER=kod] [,KEY=haslo] [,LEVEL=liczba],
```

w której z punktu widzenia ochrony dostępu do danych są istotne operandy:

USER - określa kod operatora, a w przypadku jego pominięcia pozycję opisującą "wszystkich pozostałych operatorów",

KEY - określa hasło kontrolne służące do autoryzacji dostępu operatora, sprawdzane przez MTP,

LEVEL - określa maksymalny dozwolony dla danego operatora poziom wykonywanych transakcji.

Opcja czwarta /określanie poziomu ochrony transakcji/ jest definiowana w makroinstrukcji MTPTRT, służącej do generowania tablicy transakcji. Makroinstrukcja ta ma postać:

```
MTPTRT  TYPE={INITIAL,TRANS,FINAL} [,SUFFIX=xx]
  [,NAME=nazwa] [,PROGRAM=nazwa] [,LEVEL=liczba]
  [,ENV={MTP,MPR}] [,DB={RETRIEVAL,UPDATE}]
  [,MODE={CONVERS,BATCH}] [,PRTY=liczba] ,
```

gdzie z punktu widzenia ochrony dostępu do danych są istotne operandy:

NAME - określa nazwę transakcji,

PROGRAM - określa nazwę programu związanego z daną transakcją,

LEVEL - określa poziom ochrony transakcji, a więc taką wartość poziomu dostępu operatora, która umożliwia wykonanie tej transakcji,

DB - określa, czy transakcja ma dostęp do bazy danych oraz czy będzie w niej wyszukiwać bądź ją aktualizować.

Opisane mechanizmy zabezpieczające wbudowane w MTP, umożliwiają zespołowi administratora bazy danych zaprojektowanie i wdrożenie do eksploatacji efektywnego i wszechstronnie funkcjonującego systemu ochrony dostępu do danych w części obsługującej pracę bezpośrednią /w tym wielozadaniową, wielodostępną, transakcyjną lub konwersacyjną/ w środowisku lokalnym lub tele-

komunikacyjnym. Taka praca jest podstawą funkcjonowania systemów zarządzania wspomaganego komputerem. Z tego względu wykorzystanie przez administratora możliwości MTP jest ważnym czynnikiem determinującym poprawność konstrukcji i działania tej części systemu ochrony dostępu[15]. Wprawdzie sprawy związane z zakresem i sekwencją prac projektowych, które powinien on w tym celu zrealizować, wykraczają poza ramy artykułu, lecz należy zwrócić uwagę na te akcje administratora, które decydują o stopniu przydatności wymienionych mechanizmów MTP w omawianej klasie systemów informatycznych. Do tych akcji autor zalicza:

1. Definiowanie tablic MTP, w tym zwłaszcza:

- tablicy opisującej sieć terminalową szczególnie w części opisu terminali /makroinstrukcja MTPCTT/, w której można podzielić terminale na klasy, przyznając im priorytety, przy czym najwyższy priorytet należy przyznać konsoli administratora bazy danych sterującej działanie i zamykanie MTP oraz wyprowadzającej komunikaty i dziennik pracy systemu;

- tablicy kontroli operatorów /makroinstrukcja MTPURT/, w której można nadać operatorom terminali identyfikatory i przydzielić hasła kontrolne wykorzystywane każdorazowo do inicjacji sesji; przy czym celowy jest taki podział operatorów, aby tym, którzy mają przyporządkowane najwyższe poziomy dostępu, można najczęściej zmieniać hasła;

- tablic transakcji /makroinstrukcja MTPTRT/, w których administrator może m.in. określić powiązane z poziomami dostępu operatorów poziomy ochrony transakcji. W tym celu powinien on podzielić transakcje na klasy zależne od ich funkcji /np. opisana niżej transakcja MTMT ze względu na swe możliwości powinna mieć poziom najwyższy/, a następnie za pomocą metody poziomów uprawnień[16] zdefiniować takie związki, jak poziom dostępu operatora - poziom ochrony transakcji;

---

[15] Wiąże się to z przekonaniem wielokrotnie akcentowanym przez autora, że w przypadku tak uniwersalnego narzędzia jak RODAN o poprawności rozwiązań eksploatacyjnych systemu bazy danych decyduje racjonalne, w tym często kompromisowe, wykorzystanie tkwiących w nim możliwości przez zespół administratora.

[16] Metodę poziomów uprawnień i możliwości jej wykorzystania w systemach informowania kierownictwa autor przedstawił w [5]. Por. także [1].

- tablicy inicjacji MTP /makroinstrukcja MTPTIM/, w której dzięki określeniu cykli pracy procesora komunikatów i MTP można przyspieszyć wykonywanie operacji związanych z ochroną oraz umożliwić inicjowanie różnych wersji MTP dla różnych grup operatorów i transakcji.

2. Definiowanie stron danych, czyli formatowanie ekranów /makroinstrukcje PAGE, MTPFLD oraz LINE i FIELD/,w którego czasie administrator może częściowo wspomóc działanie wymienionych zabezpieczeń i ograniczyć działania użytkownika przez stosowanie:

- wyboru z listy możliwości /tzw. menu table/,
- "formularzy dokumentów" do wypełnienia,
- komunikatów typu OUTPUT, do których programy transakcji nie mają dostępu,
- pól chronionych przed modyfikacją,
- pól niewyświetlanych /np. dla pola, do którego użytkownik wprowadza hasło kontrolne/.

3. Bezpośrednie kontrolowanie pracy użytkowników przez administratora oraz sterowanie działaniem i zamykanie pracy MTP /specjalna transakcja MTMT umożliwia uzyskiwanie informacji o aktualnym stanie systemu, zmiany parametrów inicjacji MTP, zmiany stanu terminali lub linii, sterowanie pracą procesora komunikatów itp./.

4. Wykorzystanie dziennika pracy systemu, który dzięki zawartym w nim informacjom o zdarzeniach zachodzących w środowisku MTP i MFR pozwala administratorowi identyfikować najczęstsze sposoby naruszania systemu zabezpieczeń, jego słabe miejsca i na tej podstawie prowadzić prace modernizacyjne i konserwacyjne.

Druga grupa funkcji ochrony dostępu do danych to, zgodnie z wcześniejszą specyfikacją, sterowanie dostępem według określonych dla użytkowników i elementów danych zbiorów ograniczeń dostępu. Jak wspomniano, w pracy wsadowej realizację tych funkcji zapewniają wbudowane w JOD schematu i podschematu mechanizmy blokad dostępu i współdziałające z nimi klucze dostępu wchodzące w skład podschematu, funkcji pomocniczych oraz JMD.

Klucz dostępu /privacy key/ jest to pojedyncza wartość o zdefiniowanej przez implementatora wielkości lub typie, która może być stałą,wartością zmiennej albo wynikiem działania procedury bazy danych.

Blokada dostępu /privacy lock/ jest wartością /stałą lub zmienną/ zdefiniowaną podczas opisu struktury logicznej bazy danych, porównywaną z dostarczoną przez zadanie użytkowe wartością klucza dostępu lub procedurą, która jest przywoływana i ma dostęp do właściwego klucza. Powrót procedury z wynikiem pozytywnym umożliwia dostęp do chronionych elementów danych[17].

Blokady dostępu są deklarowane w klauzulach typu PRIVACY LOCK o takim oto formacie ogólnym:

```
               [     ||funkcja COPY          ||  |
               |     ||podschematu           ||  |   {literal-1             }
               |     ||funkcje pomocnicze    ||  |   {                      }
PRIVACY LOCK   | FOR ||PFP i programu        ||  | IS{PROCEDURE procedura  }
               |     ||DISPLAY               ||  |   {-bazy-danych-1        }
               [     ||komendy JMD           ||  |
                                         [        literal-2        ]
                                         |   OR { PROCEDURE procedura } ...
                                         [      { -bazy-danych-2   }]
```

Można je ustanawiać dla dwóch różnych grup rodzajów dostępu:

1/ dla sterowania dostępem do opisów struktury logicznej i fizycznej bazy danych na poziomach:

- schematu - zakres ich działania dotyczy operacji kopiowania części lub całości schematu /COPY/ i drukowania opisu bazy danych /DISPLAY/,

- podschematu - zakres ich działania dotyczy kompilacji programów przywołujących dany podschemat /COMPILE/ i drukowania opisu części bazy ograniczonej podschematem /DISPLAY/[18].

2/ dla sterowania dostępem do elementów struktury logicznej i fizycznej bazy danych na poziomach:

- obszaru - zakres ich działania dotyczy wyłącznego lub chronionego otwarcia obszaru dla operacji wyszukiwania lub aktualizowania lub dla każdej z funkcji PFP /INTIATE, PRINT, PATCH, VERIFY/,

- zapisu - zakres ich działania dotyczy komend JMD operujących na zdefiniowanym typie zapisu /komendy: INSERT, REMOVE, STORE, DELETE, GET, MODIFY, FIND/,

---

[17] Oprócz wymienionej funkcji sterowania dostępem procedura może realizować inne działania, których przykłady autor zawarł w [2].

[18] W obecnej wersji USZBD RODAN 3.0 nie są oprogramowane blokady dla operacji wprowadzania zmian do schematu lub podschematu /ALTER/ oraz dla zmiany blokad zdefiniowanych w schemacie lub podschemacie /LOCKS/.

- zbioru bazy danych - zakres ich działania dotyczy komend JMD operujących na zdefiniowanym typie /komendy: FIND, INSERT, REMOVE/,

- zapisu podrzędnego - zakres ich działania dotyczy komend JMD operujących na wystąpieniach zapisów podrzędnych zdefiniowanego typu zbioru /komendy: FIND, INSERT, REMOVE/,

- danej elementarnej i złożonej - zakres ich działania dotyczy komend JMD operujących na zdefiniowanych danych /komendy: STORE, GET, MODIFY/[19].

Blokady dostępu ustanawiane w schemacie bazy danych dla wymienionych poziomów i zakresów mogą być zmienione w podschematach. Jest to możliwe dla poziomów: obszaru, zapisu, zbioru, zapisu podrzędnego i danej. Opisana opcja pozwala na deklarowanie odrębnych zbiorów ograniczeń dostępu dla zadań użytkowych posługujących się różnymi podschematami. W ten sposób rozszerza się elastyczność systemu ochrony dostępu, tak ważna w systemach zarządzania wspomaganego komputerem. W systemie RODAN brakuje natomiast mechanizmów selektywnie blokujących pojedyncze wystąpienia i uniemożliwiających korzystanie z nich zadaniom przetwarzanym równolegle[20].

W przypadku pracy bezpośredniej w środowisku lokalnym lub zdalnym system RODAN wykorzystuje do sterowania dostępem opisane wyżej mechanizmy blokad i kluczy dostępu. Operand DB makroinstrukcji MTPTRT umożliwia dodatkowe określenie:

- czy dana transakcja będzie działać na bazie danych,

- czy będzie ją aktualizować lub w niej wyszukiwać.

Realizację trzeciej grupy funkcji ochrony dostępu /ograniczenie możliwości wykorzystania danych w przypadku przełamania lub obejścia zdefiniowanych wyżej zabezpieczeń/ w przypadku pracy wsadowej i bezpośredniej zapewnia specyfikowana w schemacie w zadaniu RECORD specjalna procedura konwersji. Jest ona definiowana w klauzuli ENCODING/DECODING, która ma postać:

---

[19] W pracy [9] w ramach rozbudowy mechanizmów ochrony w USZBD RODAN przewiduje się opcjonalne ograniczenie zakresu działania blokad na poziomie danej elementarnej i złożonej do wartości elementów struktury /opcja WHERE klauzuli PRIVACY-dane/.

[20] Zdaniem wielu specjalistów, w tym przedstawicieli komitetu CODASYL, mechanizmy takie prowadziłyby do obniżenia parametrów eksploatacyjnych SZBD [6].

FOR{ENCODING / DECODING}[ALWAYS] CALL procedura-bazy-danych-1

Specjalna procedura konwersji przy określeniu opcji ENCODING jest wywołana podczas wykonywania komend JMD:STORE /zapisanie do bazy danych pojawienia się nowego zapisu danego typu/ i MODIFY /modyfikacja wystąpienia zapisu danego typu/. Jeśli natomiast zostaje podana opcja DECODING, wywołanie procedury konwersji jest realizowane podczas wykonania komendy GET przesyłającej wystąpienie z bazy do roboczego obszaru użytkownika. Tak funkcjonująca klauzula ENGODING/DECODING wykonuje w ograniczonym zakresie enkrypcję danych[21], a tym samym wzmacnia system ochrony dostępu do nich.

4. Wnioski

Na podstawie analizy mechanizmów zabezpieczających wbudowanych w USZBD RODAN można stwierdzić, że ich możliwości wystarczają do zaprojektowania poprawnie działającego systemu ochrony dostępu do danych w zakresie zarządzania wspomaganego komputerem. Mechanizmy systemu RODAN zapewniają bowiem ochronę elementów danych na wszystkich poziomach logicznej struktury danych przez system blokad i kluczy dostępu. Nie ma jednak możliwości selektywnego blokowania pojedynczych wystąpień, co obniża graniczną selektywność ochrony do poziomu danej elementarnej i zakresu typu elementu struktury. Można bowiem zabezpieczyć co najwyżej poszczególne dane elementarne zdefiniowane przez identyfikator bazy danych. Jednakże zgodnie ze specyfikacją wymogów, sformułowaną w punkcie 1 artykułu, w większości SZWK powinno to wystarczać. W szczególnych przypadkach przez dodatkowe zabiegi organizacyjne na etapie projektowania struktury logicznej i fizycznej bazy danych /np. podział bazy na obszary o różnych parametrach typów dostępu, podział wstępnie określonych typów zapisów na grupy wystąpień w zależności od wartości pewnych elementów danych itp./ można spełnić także wymaganie selekcji pojedynczych wystąpień lub ich grup. RODAN zapewnia również dostateczną

[21] Należy podkreślić, że w raporcie komitetu CODASYL z 1978 roku proponuje się usunięcie klauzuli ENCODING/DECODING ze specyfikacji sieciowych baz danych ze względu na różne procedury konwersji pożądane z tytułu możliwości zmian formatów danych między schematem i podschematem [6].

wnikliwość ochrony, gdyż minimalnym wspólnym kryterium dopuszczalności jest typ operacji. Wnikliwość dotyczy więc poszczególnych komend JMD, funkcji Procesora Funkcji Pomocniczych i niezależnego programu pomocniczego DISPLAY. System RODAN umożliwia dodatkowo stosowanie enkrypcji danych za pomocą specjalnej procedury konwersji. Jeżeli chodzi o wymogi nie związane bezpośrednio z ochroną dostępu do danych, to należy wspomnieć, że w skład USZBD RODAN wchodzą programy pomocnicze /niezależne i wchodzące w skład PFP/ pozwalające na prowadzenie działalności audytorskiej. Możliwe jest również stosowanie dwóch rodzajów odtwarzania stanów bazy danych: automatycznego odtwarzania krótkiego i sterowanego manualnie odtwarzania długiego[22].

Użytkownicy systemu RODAN są identyfikowani. Szczególne możliwości w tej mierze zapewnia MTP. Ich działania są autoryzowane dzięki rozwiniętemu systemowi sterowania dostępem /blokady i klucze dostępu, poziomy dostępu i poziomy ochrony w MTP/. Możliwe jest również monitorowanie ich pracy, zarówno bieżące, jak i za pomocą dzienników pracy systemu, a także bezpośrednie interweniowanie w przypadku podejmowania przez użytkowników działań niezgodnych z przyznanymi im uprawnieniami.

Można więc przyjąć, że spełnione są sformułowane wcześniej wymagania rozpatrywanej klasy systemów, zarówno w przypadku pracy wsadowej, jak i bezpośredniej, z takimi atrybutami, jak wielozadaniowość, wielodostęp i teleprzetwarzanie.

---

[22] Szczegółowy opis i analiza możliwości odtwarzania w USZBD RODAN przedstawiono w [8].

LITERATURA

[1] B. D a v i s, The Selection of Database Software, Manchester 1977, NCC Publications.

[2] M. D y c z k o w s k i, Mechanizmy ochrony dostępu do danych w wybranych systemach zarządzania bazą danych, w: Informatyka. Rachunkowość, Prace Naukowe Akademii Ekonomicznej, nr 181, Wrocław 1981.

[3] M. D y c z k o w s k i, Metoda poziomów uprawnień jako środek selektywnej ochrony dostępu do danych w systemie informowania kierownictwa, w: Informatyka. Wybrane zagadnienia, Prace Naukowe Akademii Ekonomicznej, nr 205, Wrocław 1982.

[4] M. D y c z k o w s k i, Zagrożenia ochrony dostępu do danych w systemach zarządzania wspomaganego komputerem, [w:] Informatyka. Teoria. Praktyka, Prace Naukowe Akademii Ekonomicznej, nr 235, Wrocław 1983.

[5] M. D y c z k o w s k i, M. S k w a r n i k, Identyfikacja użytkowników w systemach nauczania wspomaganego komputerem. Materiały konferencyjne INFOGRYF'80, TNOiK i Politechnika Szczecińska, Szczecin 1980.

[6] L. M a c i a s z e k, Baza danych w systemie wielodostępnym. Specyfika technologii. Materiały konferencyjne. Wielodostęp w zarządzaniu, badaniach naukowych i dydaktyce, TNOiK, Poznań 1980.

[7] J. M a r t i n, Principles of Data-Base Management, Englewood Cliffs 1976, Prentice-Hall Inc.

[8] Przegląd metod zarządzania zbiorami i terminami dla potrzeb systemu działającego w czasie rzeczywistym na R-32. ZNB Akademii Ekonomicznej we Wrocławiu, 1981 /raport badawczy/.

[9] W. S t a n i s z k i s, Mechanizmy ochrony danych w systemach zarządzania bazą danych, Warszawa 1978, OBRI.

[10] Uniwersalny System Zarządzania Bazą Danych RODAN. Dokumentacja użytkowa części I-IV, Warszawa 1979, CPiZI.

ANALYSIS OF THE USEFULNESS OF THE MEANS OF ACCESS PROTECTION RODAN FOR THE COMPUTER-AIDED MANAGERIAL SYSTEM

Summary

The article attempts to determine the degree of usefulness of the means of access protection /in the RODAN system/ for the computer-aided managerial system. Reasons for and principles of access protection in computer-aided management are presented here. An analysis of access protection in the RODAN system is also carried out.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Colette R o l l a n d
Uniwersytet Paris I Panthéon-Sorbonne
Jan S t ę p n i e w s k i
Akademia Ekonomiczna we Wrocławiu

METODA KONSTRUKCJI RELACYJNEGO MODELU DANYCH NA PRZYKŁADZIE RACHUNKOWOŚCI FINANSOWEJ

Część II. Model dynamiki systemów*

1. Wprowadzenie

W pierwszej części naszego artykułu [5] przedstawiliśmy relacyjny model danych rachunkowości finansowej w aspekcie statycznym, tzn. składniki systemu rzeczywistości i ich strukturę. W celu wiernego odzwierciedlenia rzeczywistych cech rachunkowości finansowej oraz doskonalenia koncepcji systemu informatycznego rachunkowości zaproponujemy koncepcję wszechstronnego przedstawienia wszystkich aspektów rzeczywistości [7]. Nie można się ograniczać tylko do przedstawiania statycznego. Trzeba także uwzględnić aspekt dynamiczny, odpowiednio do rozwoju systemu w miarę upływu czasu.

Model koncepcyjny dynamiki systemów był rozwinięty i sprawdzony w dziedzinie systemów informacyjnych zarządzania w ostatnich trzech latach przez zespół naukowy Uniwersytetu Paris I Panthéon-Sorbonne [1-4].

2. Podstawowe kategorie pojęć dynamiki systemów

Schemat dynamiki systemów stanowił rezultat procesu modelowania struktury systemu rzeczywistego za pomocą trzech pojęć: klasa obiektów /K-OBIEKT/, klasa operacji /K-OPERACJA/ i klasa zdarzeń /K-ZDARZENIE/. Wymienione pojęcia zdefiniowano [4] w

*Część I opublikowana została w zeszycie: Wspomagane komputerem tworzenie systemów informatycznych, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 228, Wrocław 1984.

formalizmie relacyjnym przez trzy odpowiadające im typy relacji w formie znormalizowanej, gwarantującej minimalizację schematu koncepcyjnego.

K-OBIEKT

Pojęcie k-obiektu pozwala na odzwierciedlenie elementów lub istoty systemu rzeczywistości i ich struktury. K-obiekt jest reprezentacją aspektu czasowego pewnej klasy istoty rzeczywistości. Pojęcie k-obiektu jest pochodne od pojęcia relacji w 3 formie znormalizowanej, z tym że dodatkowo uwzględnia ono czas i swoją formę. K-obiekt przedstawia stan elementarny systemu rzeczywistego.

K-OPERACJA

Pojęcie k-operacji pozwala odzwierciedlić transformację systemu rzeczywistego. K-operacja jest zdefiniowana w odniesieniu do k-obiektu. Jest to reprezentacja pewnej klasy operacji systemu rzeczywistego, która modyfikuje stany istoty danej klasy obiektów. Zaistnienie danej k-operacji modyfikuje stan pojedynczego wystąpienia obiektu. Jedna k-operacja modyfikuje jeden i tylko jeden obiekt. K-operacja reprezentuje transformację elementarną systemu rzeczywistości.

K-ZDARZENIE

Pojęcie k-zdarzenia pozwala na odzwierciedlenie szczegółowych zmian stanu systemu rzeczywistego. K-zdarzenie jest definiowane w odniesieniu do k-operacji i k-obiektu. Jest to reprezentowanie pewnej klasy zmian stanu istoty rzeczywistości, które rozpoczynają operacje. Określając dokładniej, zaistnienie k-zdarzenia zmienia stan danego wystąpienia k-obiektu, które powoduje zajście jednej lub wielu operacji. Jedno k-zdarzenie odpowiada jednej zmianie stanu jednego k-obiektu.

K-zdarzenie reprezentuje elementarną zmianę stanu systemu, która z kolei jest początkiem jednej lub licznych transformacji elementarnych. Rozpoczęcie transformacji może być warunkowe lub iteracyjne.

Zilustrujmy trzy opisane kategorie pojęć dynamiki systemów na przykładzie rachunkowości finansowej /kontynuacja przykładu z cz. I/.

K-OBIEKTY

OPG - operacja gospodarcza
KTPRZ - konto przedmiotowe
KTP - konto podmiotowe
KTSYN - konto syntetyczne
STAN-KTPRZ - stan konta przedmiotowego
STAN-KTP - stan konta podmiotowego
STAN-KTSYN - stan konta syntetycznego
BILANS - bilans roczny lub na żądanie
KAL - kalendarz zamykania kont i wykazywania stanów

K-OPERACJE

OP1 - otwarcie konta przedmiotowego
OP2 - otwarcie konta podmiotowego
OP3 - otwarcie konta syntetycznego
OP4 - modyfikacja stanu konta przedmiotowego
OP5 - modyfikacja stanu konta podmiotowego
OP6 - modyfikacja stanu konta syntetycznego
OP7 - zamknięcie konta przedmiotowego
OP8 - zamknięcie konta syntetycznego
OP9 - sporządzenie bilansu rocznego lub na żądanie

K-ZDARZENIA

ZD1 - wystąpienie danej operacji gospodarczej
ZD2 - wystąpienie momentu manipulacji na koncie według przyjętego kalendarza
ZD3 - wystąpienie końca roku gospodarczego

W celu ułatwienia rozważań przed przystąpieniem do modelowania dynamiki systemu rachunkowości finansowej przypomnimy schemat koncepcyjny w ujęciu statycznym przedstawiony w cz. I [5]:

KTSYN /NRKTSYN,NZWKTSYN/
KTPRZ /NRKTPRZ,NRKTSYN,NZWPRZ/
KTP /NRKTP,NZWP,ADRP/
OPG /NROP,DATOP,TROP,WARTOP,ROP,NKTP,NRKTPRZ/
KTPRZ-KTP /NRKTPRZ,NRKTP/
STAN-KTP /DATKTP,NRKTP,OBRWN,OBRMA/

STAN-KTPRZ /DATKTPRZ,NRKTPRZ,OBRWN,OBRMA/
STAN-KTSYN /DATKTSYN,NRKTSYN,OBRWN,CBRMA/
BILANS /DATBIL,NRBIL/
KAL /DAT,NRKT/

3. Schemat koncepcyjny dynamiki systemu rachunkowości finansowej

Schemat koncepcyjny jest pewnym zbiorem relacji opierających się na trzech typach: k-obiekt, k-operacja i k-zdarzenie. Zawiera on dwa subschematy odpowiadające aspektom statycznym i dynamicznym modelowanego systemu rzeczywistości [4].

Subschemat k-obiektów, lub inaczej statyczny, jest odzwierciedleniem struktury elementów systemu fizycznego.

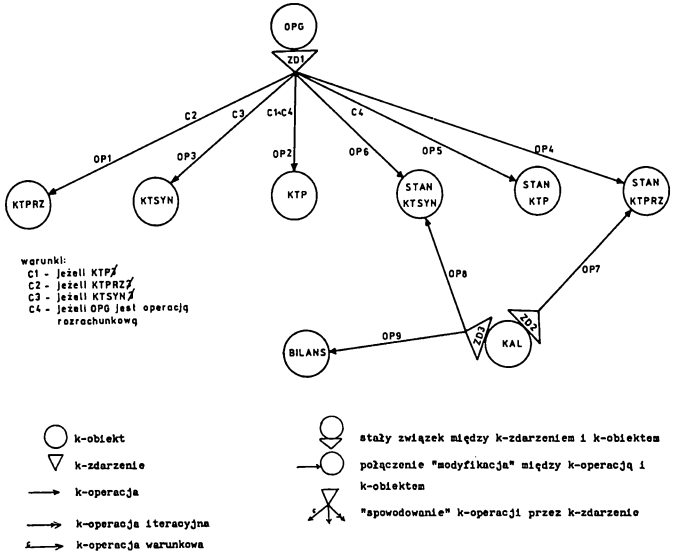
Subschemat k-operacji i k-zdarzeń, lub inaczej dynamiczny, modeluje strukturę zawartości systemu fizycznego. Otrzymany model prowadzi do przedstawienia dynamiki typu przyczynowego, gdzie zdarzenia powodują operacje, te zaś modyfikują obiekty. Pewne zmiany stanu obiektów stają się z kolei zdarzeniami. Na schemacie dynamika jest reprezentowana przez wzajemne powiązanie trzech pojęć, reprezentowanych przez trzy rodzaje symboli graficznych [2], jak to pokazano na rys. 1.

Wystąpienie k-obiektu OFG jest to zdarzenie ZD1, które wywołuje operację OP1 /otwarcie odpowiedniego konta przedmiotowego/, jeżeli warunek C2 /dotychczas to konto nie istniało/ jest prawdziwy, i modyfikację stanu konta przedmiotowego /OP4/.

Jeżeli warunki C1 /konto podmiotowe dotychczas nie istniało/ oraz C4 /operacja ekonomiczna jest operacją rozrachunkową/ są prawdziwe, to zdarzenie ZD1 powoduje operację OP2 /otwarcie konta podmiotowego/. Jeżeli warunek C4 jest prawdziwy, zdarzenie ZD1 rozpoczyna operację OP5 /modyfikacja stanu konta podmiotowego/. Jeżeli warunek C3 jest prawdziwy, zdarzenie ZD1 rozpoczyna operację OP3 /otwarcie konta syntetycznego/ i operację OP6 /modyfikacja stanu konta syntetycznego w każdym przypadku/.

ZD2 - wystąpienie momentu manipulacji na koncie według przyjętego kalendarza. Jest to zdarzenie, które rozpoczyna operację OP7 /zamknięcie kont przedmiotowych/.

ZD3 - wystąpienie końca roku gospodarczego. Jest to zdarzenie, które powoduje operację OP8 /zamknięcie kont syntetycznych/ i operację OP9 /sporządzenie rocznego bilansu lub na żądanie/.



Rys. 1. Koncepcja dynamiki systemu rachunkowości finansowej

4. Wnioski

Dotychczasowe doświadczenia w realizacji tej metody pozwoliły na wyciągnięcie następujących wniosków:

- przedstawiony model jest prosty i efektywny. Nadaje się zwłaszcza do analizy i modyfikacji systemów informatycznych funkcjonujących w czasie rzeczywistym. Definicje semantyczne pojęć i ich elementarność to dwa korzystne aspekty modelu, ułatwiające opracowanie koncepcji i wprowadzenie pewnego rygoru metodologicznego,

- do systematycznej eksploatacji metody niezbędne są narzędzia z zakresu programowania, służące do weryfikacji spójności, symulacji logiki połączenia zdarzeń, oraz dokumentacja.

- w modelowaniu dynamiki systemów informatycznych nieuniknione jest jednoczesne występowanie zdarzeń i operacji wywołanych przez te zdarzenia. Koncepcja rozwiązania tego problemu wydaje się bardzo interesująca, jest sprawą przyszłości,

- na podstawie wykorzystanego w części I i II przykładu z rachunkowości finansowej można potwierdzić wyjątkową predyspozycyjność systemu rachunkowości do adaptacji relacyjnego modelu danych [6].

LITERATURA

[1] C. R o l l a n d, La conception des systèmes de bureau, Actes du Congrès AFCET 1979, Grenoble.

[2] C. R o l l a n d, A méthodology for information systems design, Proceeding of the NCC 81, Chicago 1981.

[3] C. R o l l a n d, Le projet Remora, Rapport final de contrat INRIA /SESORI, Paris 1979.

[4] C. R o l l a n d, S. L e i f e r t, C. R i c h a r d, Toods for information system dynamics management. Proceedings of the 5th international conference on VLDB, Rio de Janeiro 1979.

[5] C. R o l l a n d, J. S t ę p n i e w s k i, Metoda konstrukcji relacyjnego modelu danych na przykładzie rachunkowości finansowej, cz. I, [w:] Informatyka. Wybrane zagadnienia, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 213, Wrocław 1982.

[6] J. S t ę p n i e w s k i, Le rôle de la comptabilité dans la construction de systeme informatique de gestion integré, "Revue Française de comptabilité", 1982, nr 127.

[7] J. S t ę p n i e w s k i, C. R o l l a n d, Une methode d'integration de système informatique de gestion. Actes du Seminaire AFCET, INFORSID-82, Toulouse 1982.

A METHOD FOR CONSTRUCTING A RELATIONAL DATA MODEL AS EXEMPLIFIED IN ACCOUNTANCY

Summary

The systems dynamics scheme was developed in the course of a real system structure simulation with three categories involved in the process: the classes of objects, operations, and events. The above stated concepts were defined formally, by means of three corresponding types of relations in a standard form which minimalizes the systems dynamic conceptual scheme.

The model presented here is noteworthy for its simplicity and effectiveness. It is especially suitable for analyzing and modifying real time informatic systems. Semantic definitions with their elementary character are two advantageous aspects of the model. They simplify. the work involved in arriving at the conception as well as in introducing some methodological rigour.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Stanisław Z i m n o c h o
Akademia Ekonomiczna we Wrocławiu

ZASTOSOWANIE INTERPOLACJI DO PRZESZUKIWANIA ZBIORÓW DANYCH

1. Informacje wstępne

Interpolacją nazywamy metody wyznaczania wartości funkcji odpowiadających pośrednim, nieznanym wartościom argumentu w dowolnym punkcie przedziału (x_0,x_n), gdy dane są wartości funkcji w punktach $x_0, x_1, x_2, x_3 \ldots\ldots x_n$. Na temat interpolacji szczegółowo pisano w literaturze dotyczącej metod numerycznych oraz metod opracowywania wyników doświadczeń [2, s. 43-63; 8, s. 107-117 .

Pierwszą pracą, w której zaproponowano wykorzystanie idei interpolacji do przeszukiwania zbiorów danych w pamięci o dostępie bezpośrednim, jest artykuł W.W. Petersona [16]. Autor na podstawie analizy algorytmu stwierdził, że w przypadku jednostajnego rozkładu wartości identyfikatora dolną granicą średniej liczby dostępów do zbioru w przeszukiwaniu interpolacyjnym jest $\frac{1}{2}\log_2 N$, gdzie N oznacza liczebność zbioru identyfikatorów. W późniejszych pracach innych autorów dowiedziono, że istnieje błąd w tym oszacowaniu. [3;15;20].

Pobieżny opis interpolacji bez podania szczegółów oraz bez analizy algorytmu zawierają prace: D. Knutha [10, s. 416-419], C. Price'a [17] oraz H.S. Kruijera [13]. Dokładniejszą analizę podają A.C. Yao i F.F. Yao w [20]. Autorzy za pomocą skomplikowanego kombinatorycznego dowodu wykazali, że średnia liczba dostępów dla tego algorytmu, w przypadku jednostajności rozkładu identyfikatorów, wynosi $\log_2\log_2 N$. Kolejnymi, opublikowanymi

pracami poświęconymi analizie algorytmu interpolacyjnego są: dysertacja G.H. Gonneta [3] oraz artykuły Y. Perl'a, A. Itai, H. Avni'ego [15] i G.H. Gonneta, L.D. Rogersa, J.A. George'a [5]. We wszystkich wymienionych opracowaniach w analizie algorytmu zakłada się, że przeszukiwany zbiór identyfikatorów ma rozkład jednostajny. Brakuje natomiast prób oceny własności tej metody przy bardziej skomplikowanej charakterystyce rozkładu wartości identyfikatora, jaka zwykle cechuje istniejące zbiory danych w systemach informatycznych.

2. Opis algorytmu INTLIN

Przeszukiwanie interpolacyjne jest algorytmem, który może być wykorzystywany do zbiorów danych znajdujących się w pamięci o dostępie bezpośrednim. Wówczas powinny być spełnione następujące warunki:

- zbiór musi być uporządkowany w rosnącej lub malejącej sekwencji wartości identyfikatora[1],
- względny adres i-tego zapisu w zbiorze można wyznaczyć jednoznacznie,
- przed rozpoczęciem przeszukiwania znane są: najmniejsza i największa wartość identyfikatora w zbiorze.

Wymienione warunki są spełnione wówczas, gdy zbiór ma strukturę wektora:

$$(Z_0, Z_1, Z_2, \ldots, Z_N), \qquad /1/$$

przy czym elementy tego wektora są uporządkowanymi zapisami stałej długości lub tablicą adresową, nadającą porządek zapisom stałej lub zmiennej długości. Porządek w zbiorze jest wyznaczony zależnością:

$$T_i < T_{i+1}, \quad i=0,1, \ldots, N-1, \qquad /2/$$

gdzie: T_i - wartość identyfikatora i-tego zapisu[2].
Zatem najmniejszą wartością identyfikatora w zbiorze jest T_0, a największą T_N.

---

[1] W dalszych rozważaniach przyjmujemy założenie o rosnącym uporządkowaniu zbioru.

[2] Dalsza analiza algorytmu będzie dotyczyć wyłącznie identyfikatorów bez pozostałej części zapisów.

Względny adres i-tego zapisu można wyznaczyć za pomocą wzoru:

$$adr\ (Z_1) = i \cdot f, \qquad /3/$$

gdzie f - długość zapisu[3].

Możemy więc przedstawić zbiór identyfikatorów jako funkcję monotonicznie rosnącą, odwzorowującą podzbiór liczb naturalnych w zbiór liczb rzeczywistych:

$$f : I \longrightarrow R . \qquad I = (0,1,2, \ldots , N) \qquad /4/$$

oraz

$$f(i) < f(i+1) , \quad i \in I . \qquad /5/$$

Metoda przeszukiwania interpolacyjnego polega na iteracyjnym wyznaczaniu indeksu cięcia /ang. "cut index"/[4], to znaczy określaniu za pomocą funkcji interpolującej pozycji takiego zapisu w zbiorze, który - w razie założenia określonego rozkładu wartości identyfikatora - powinien być identyfikatorem szukanym[5]. Argumentami funkcji interpolującej, oprócz identyfikatora szukanego K, są:

- indeks id największego znanego identyfikatora mniejszego od identyfikatora szukanego

$$T_o \leqslant T_{id} < K , \qquad /6/$$

- indeks ig najmniejszego znanego identyfikatora większego od identyfikatora szukanego

$$K < T_{ig} \leqslant T_N , \qquad /7/$$

- liczba identyfikatorów RO wyznaczana według wzoru:

$$RO = ig - id. \qquad /8/$$

Z wzoru na interpolację liniową i zgodnie z regułami interpolacji odwrotnej otrzymujemy wzór na funkcję interpolującą:

$$ic = \frac{K-T_{ig}}{T_{id}-T_{ig}}\ id + \frac{K-T_{id}}{T_{ig}-T_{id}}\ ig. \qquad /9/$$

---

[3] Wzór /3/ dotyczy zbioru zapisów stałej długości. Dla zapisów zmiennej długości z tablicą adresową algorytm wyznaczania adresu jest inny /por. [22].

[4] Indeks cięcia będzie oznaczany symbolem ic.

[5] Założenie o jednostajnym rozkładzie wartości identyfikatora implikuje wykorzystanie wzoru na interpolację liniową.

Z wzoru /8/ mamy: ig=RO-id i stąd:

$$ic' = \frac{K-T_{ig}}{T_{id}-T_{ig}}id + \frac{K-T_{id}}{T_{ig}-T_{id}}(RO+id), \qquad /10/$$

co po odpowiednich przekształceniach daje:

$$ic` = \left(\frac{K - T_{id}}{T_{ig} - T_{id}} \cdot RO\right) + id. \qquad /11/$$

Ponieważ wartości ic muszą być liczbami całkowitymi, przyjmujemy:

$$ic = \left[ic` \right] = \left\lfloor \left(\frac{K-T_{id}}{T_{ig}- T_{id}} RO\right) \right\rfloor + id \Bigg\rfloor . \qquad /12/$$

Zatem, korzystając z podanych wzorów, algorytm przeszukiwania interpolacyjnego INTLIN możemy opisać następująco:

1° Jeżeli $K < T_o$ lub $K > T_N$,
to szukanie niepomyślne /brak szukanego identyfikatora w zbiorze/.
przejdź do 12°:

2° jeżeli $K = T_o$,
to szukanie pomyślne (0),
przejdź do 12°;

3° jeżeli $K = T_N$,
to szukanie pomyślne (N),
przejdź do 12°;

4° id:=0:
ig:= N;

5° RO:= ig-id:

6° jeżeli ig-id $\leqslant$ 1,
to szukanie niepomyślne.
przejdź do 12°;

7° $ic := \left\lfloor \left(\frac{K - T_{id}}{T_{ig} - T_{id}} RO\right) + id \right\rfloor$;

8° jeżeli ic = ig, to ic:= ic-1;

9° jeżeli $K = T_{ic}$,
to szukanie pomyślne (ic).
przejdź do 12°;

10° jeżeli $K < T_{ic}$, to ig:= ic.
przejdź do 5°;
11° jeżeli $K > T_{ic}$, to id:= ic,
przejdź do 5°;
12° KONIEC.

W razie znalezienia danego identyfikatora - w opisie algorytmu podaje się jego indeks /numer kolejny w zbiorze/.

3. Funkcjonowanie algorytmu INTLIN

Budowa funkcji interpolującej /12/ zakłada liniowy rozkład wartości identyfikatora w zbiorze, to znaczy, że spełniony jest warunek:

$$T_{i+1} - T_i = c \ , \quad i = 0,1,2, \ldots , N-1, \qquad /13/$$

gdzie c - stała[6]. Gdyby się jednak przyjęło, że przeszukiwany zbiór będzie miał tę własność, to konstrukcja całego algorytmu byłaby dużo prostsza. Odszukanie każdego istniejącego identyfikatora wymagałoby dokładnie jednego dostępu do zbioru:

$$ic = \frac{K - T_o}{T_N - T_o} \cdot N \ . \qquad /14/$$

Brak natomiast identyfikatora w zbiorze można by stwierdzić bez przeszukiwania. Jeżeli nie istnieje takie całkowite k, że:

$$k \cdot c = K, \qquad /15/$$

gdzie c jest zdefiniowane wzorem /13/, to szukany identyfikator w zbiorze nie występuje.

Spełnienie przez zbiór identyfikatorów warunku /13/ należy jednak uznać za przypadek szczególny w komputerowym przetwarzaniu danych[7]. Bardziej realistycznym założeniem będzie stwierdzenie, że zbiór identyfikatorów ma rozkład probabilistyczny. W takiej sytuacji konieczne staje się rozbudowanie algorytmu do postaci wcześniej przedstawionej.

---

[6] Jest to założenie o deterministycznym charakterze rozkładu wartości identyfikatora.

[7] Przy liniowym lub zbliżonym do liniowego rozkładzie wartości identyfikatora projektowane są zbiory o organizacji bezpośredniej z wykorzystaniem transformacji interpolacyjnej do generowania adresów zapisów w zbiorze, por. [11;12;19].

Przeszukiwanie przebiega iteracyjnie. Przed pierwszym dostępem do zbioru "indeks dolny" /id/ oraz "indeks górny" /ig/ znajdują się na krańcach zbioru. W trakcie wykonywania kolejnych iteracji obszar, w którym powinien się znajdować szukany identyfikator, zawęża się aż do momentu odszukania zapisu lub do momentu, w którym zbiór identyfikatorów pomiędzy ig a id stanie się pusty.

W dostępnych publikacjach poświęconych problematyce przeszukiwania zbiorów danych, można znaleźć dokładną analizę procesu przeszukiwania interpolacyjnego zbiorów identyfikatorów o rozkładzie jednostajnym. Przy takim założeniu indeks cięcia jest zawsze oczekiwanym miejscem szukanego zapisu. Zatem większość dostępów odbywa się na stosunkowo małym obszarze, w sąsiedztwie szukanego zapisu. W środowisku stronicowanym, w którym pewna liczba kolejnych zapisów znajduje się na pojedynczej stronie, dzięki tej właściwości algorytmu znacznie maleje liczba przesłań danych między pamięcią operacyjną a pamięcią zewnętrzną[8].

Z analizy algorytmu, przeprowadzonej między innymi przez G.H. Gonneta w [3] oraz [5], a także przez innych autorów, wynika, że dla identyfikatorów o rozkładzie jednostajnym średnia liczba dostępów, zarówno dla szukania pomyślnego, jak i niepomyślnego, wyraża się wzorem:

$$DL_1(N) = \log_2\log_2 N. \qquad /16/$$

Ponieważ funkcja $\log_2\log_2 x$ jest bardzo płaska /por. rys. 1/, w praktyce liczba 5 może być uznana za górną granicę średniej liczby dostępów, gdyż nie istnieją obecnie na świecie zbiory danych liczące więcej niż $2^{32} = 4.294.967.296$ zapisów[9]. Po uwzględnieniu wszystkich wymienionych własności przeszukiwania interpolacyjnego należy stwierdzić, że w określonych warunkach algorytm ten może konkurować z przeszukiwaniem binarnym, a nawet z trzypoziomowym przeszukiwaniem indeksowo-sekwencyjnym [1;15].

---

[8] Szczegółowa analiza wpływu blokowania zapisów na efektywność algorytmu znajduje się w [21].

[9] $\log_2\log_2 2^{32} = 5$.

4. Modyfikacje algorytmu interpolacyjnego

Opisany klasyczny algorytm interpolacyjny ma, w porównaniu z innymi algorytmami przeszukiwania zbiorów danych, stosunkowo mocno skomplikowaną strukturę. Jednocześnie charakteryzuje się tym, że dla jednostajnego lub zbliżonego do jednostajnego rozkładu wartości identyfikatora kolejne dostępy są realizowane w pobliżu identyfikatora szukanego. Te dwa czynniki spowodowały wysunięcie koncepcji uproszczenia algorytmu interpolacyjnego, polegającego na tym, że tylko pierwszy dostęp jest realizowany zgodnie z wartością funkcji interpolującej, wszystkie następne natomiast odbywają się sekwencyjnie, zgodnie z kolejnością występowania zapisów w zbiorze. Ten wariant algorytmu po raz pierwszy opisał C.E. Price w [17]. W tym opracowaniu brakuje jednak dokładnego opisu algorytmu oraz określenia jego efektywności dla rozkładów wartości identyfikatora. Pełną analizę algorytmu w warunkach rozkładu jednostajnego zawiera artykuł G.H.Gonneta i L.D. Rogersa [4]. Autorzy, oprócz oszacowania średniej liczby dostępów, przedstawili analizę efektywności w zbiorach posiadających zapisy blokowane.

Przyjmując oznaczenia definiowane wzorami /6/ - /18/ algorytm INTSEK opisujemy następująco:

1° Jeżeli $K < T_o$ lub $K > T_N$,
to szukanie niepomyślne /brak szukanego identyfikatora w zbiorze/,
przejdź do 9°;

2° jeżeli $K = T_o$,
to szukanie pomyślne (0),
przejdź do 9°;

3° jeżeli $K = T_N$,
to szukanie pomyślne (N),
przejdź do 9°.

4° $ic = \left\lceil \frac{K-T_o}{T_N - T_o} \cdot N \right\rceil$;

5° jeżeli ic = N, to ic:=ic-1;

6° jeżeli $K = T_{ic}$,
to szukanie pomyślne (ic),
przejdź do 9°;

7° jeżeli $K < T_{ic}$, to:
  7a° ic:=ic-1;
  7b° jeżeli $K=T_{ic}$, to szukanie pomyślne (ic);
  7c° jeżeli $K > T_{ic}$, to szukanie niepomyślne,
      przejdź do 9°;
  7d° przejdź do 7a°;
8° jeżeli $K > T_{ic}$, to:
  8a° ic:=ic+1;
  8b° jeżeli $K=T_{ic}$, to szukanie pomyślne (ic);
  8c° jeżeli $K < T_{ic}$, to szukanie niepomyślne,
      przejdź do 9°;
  8d° przejdź do 8a°;
9° KONIEC.

Przeszukiwanie za pomocą algorytmu INTSEK, podobnie jak za pomocą algorytmu INTLIN, przebiega iteracyjnie. Istnieje jednak zasadnicza różnica w sposobie wyznaczania kolejnych indeksów cięcia dla obydwu algorytmów. Metoda przyjęta w algorytmie INTSEK odznacza się dużą prostotą i łatwością oprogramowania. Ma ona jednak poważną wadę. Mianowicie, w razie dużych błędów interpolacji w kroku 4° algorytmu liczba dostępów będzie również zbyt duża. Wówczas stosowanie algorytmu przestaje być uzasadnione.

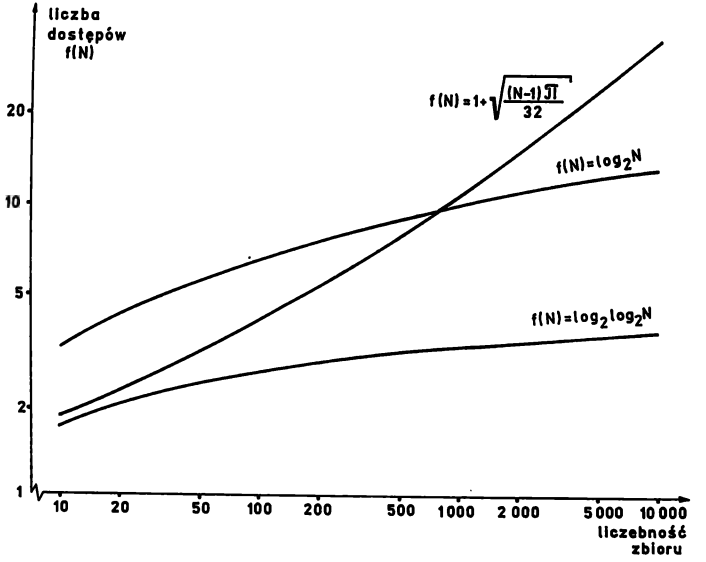
G.H. Gonnet i L.D. Rogers w [4] wykazali, że przeciętna liczba dostępów dla zbioru identyfikatorów o rozkładzie jednostajnym wynosi:

$$DL_B(N) = 1 + \sqrt{\frac{(N-1)\,\pi}{32}} + O\left(\frac{1}{\sqrt{N}}\right). \qquad /17/$$

Przebieg funkcji określonej wzorem /17/ przedstawia rys. 1. Jej kształt jest mniej płaski niż funkcji logarytmicznej. Dlatego zastosowanie algorytmu INTSEK może dać lepsze rezultaty niż przeszukiwanie binarne jedynie dla zbiorów mało licznych /mających mniej niż 500 zapisów/ lub zblokowanych[10].

Kolejną, zmodyfikowaną wersję algorytmu, wykorzystującą ideę interpolacji, stanowi algorytm interpolacyjno-skokowy zaproponowany przez Y. Perla i E.M. Reingolda w [14]. Nowością

[10] Analiza algorytmu w warunkach blokowania zapisów znajduje się w [21].



Rys. 1. Przebieg funkcji określających średnią liczbę dostępów przy poszukiwaniu binarnym, interpolacyjnym oraz interpolacyjno-skokowym

w algorytmie jest to, że pierwszy dostęp do zbioru jest realizowany za pomocą funkcji interpolującej, podobnie jak w algorytmie INTSEK, w kolejnych natomiast nie uwzględnia się wartości identyfikatorów na krańcach przeszukiwanego podzbioru. Wartość indeksu cięcia zależy od różnicy między indeksami: dolnym /id/ oraz górnym /ig/, znajdującymi się na krańcach przeszukiwanego podzbioru, i wyraża się wzorem:

$$ic = ic \pm sk, \qquad /18/$$

gdzie:

$$sk = \left\lceil \sqrt{ig - id} \right\rceil . \qquad /19/$$

Zatem formalny opis algorytmu INTPRW przedstawia się następująco:

1° Jeżeli $K < T_o$ lub $K > T_N$,
to szukanie niepomyślne (brak szukanego identyfikatora w zbiorze),
przejdź do 10°;

2° jeżeli $K = T_o$, to szukanie pomyślne (0),
przejdź do 10°;

3° jeżeli $K = T_N$, to szukanie pomyślne (N),
przejdź do 10°;

4° id:= 0;
ig:= N;

5° $ic = \left\lfloor \frac{K - T_o}{T_N - T_o} \cdot N \right\rfloor$;

6° jeżeli ic = N, to ic:= ic-1;

7° jeżeli $K = T_{ic}$, to szukanie pomyślne (ic),
przejdź do 10°;

8° jeżeli $K < T_{ic}$, to ig:= ic
oraz wykonaj 8a° - 8g°;

8a° $sk = \left\lceil \sqrt{ig-id} \right\rceil$;

8b° jeżeli ig-id≤1, to szukanie niepomyślne (brak szukanego identyfikatora w zbiorze),
przejdź do 10°;

8c° ic := max (ic - sk, id);

8d° jeżeli ic = id, to przejdź do 9°;

8e° jeżeli $K = T_{ic}$, to szukanie pomyślne (ic),
przejdź do 10°;

8f° jeżeli $K < T_{ic}$, to ig := ic oraz przejdź do 8b°;
8g° jeżeli $K > T_{ic}$, to przejdź do 9°;
9° jeżeli $K > T_{ic}$, to id := ic oraz
wykonaj 9a° - 9g°;
9a° sk = $\lceil\sqrt{ig - id}\rceil$;
9b° jeżeli ig - id$\leqslant$1, to szukanie niepomyślne (brak szukanego identyfikatora w zbiorze),
przejdź do 10°;
9c° ic := min (ic + sk, ig);
9d° jeżeli ic = ig, to przejdź do 8°;
9e° jeżeli $K = T_{ic}$, to szukanie pomyślne (ic),
przejdź do 10°;
9f° jeżeli $K > T_{ic}$, to id := ic oraz przejdź do 9b°;
9g° jeżeli $K < T_{ic}$, to przejdź do 8°;
10° KONIEC.

Powyższy algorytm, podobnie jak prezentowane poprzednio, funkcjonuje iteracyjnie. Kolejne indeksy cięcia wyznaczane są w odstępach, których wielkość zależy od dystansu między indeksem dolnym a indeksem górnym. Dlatego w razie dużej wartości wyrażenia sk (wzór /19/) następuje znacznie szybsze, niż w przypadku algorytmu INTSEK, lokalizowanie szukanego identyfikatora. Również przy bardzo niekorzystnym rozkładzie wartości identyfikatora, gdy algorytm INTLIN będzie wymagać dużej liczby dostępów[11] powyższa własność algorytmu INTPRW czyni go najbardziej efektywnym spośród trzech dotychczas opisanych. Y. Pearl i E.M. Reingold wykazali, że dla jednostajnego rozkładu wartości identyfikatora przeciętna liczba dostępów jest funkcją N i wyraża się wzorem:

$$DL_F(N) = 2.027029 \log_2\log_2 N \approx 2 \log_2\log_2 N, \qquad /20/$$

konieczna natomiast liczba dostępów w najbardziej niekorzystnym przypadku jest równa:

$$DL_{pmax}(N) = \sqrt{N} + 0\sqrt[4]{N}. \qquad /21/$$

Zestawienie wartości wyrażeń /20/, /21/ oraz funkcji $\log_2 x$ zawiera tabela 1.

---

[11] Maksymalna, możliwa liczba dostępów, zarówno dla algorytmu INTLIN, jak i algorytmu INTSEK, wynosi N /por. [4;5;14]/.

Tabela 1

Zestawienie wartości funkcji określonych wzorami /20/ i /21/

N	$\log_2 N$	$2 \log_2 \log_2 N$	$\sqrt{N} + \sqrt[4]{N}$
4	2	2	3,4
16	4	4	6
64	6	5.170	10.8
256	8	6	20
1024	10	6.644	37,7
4096	12	7.170	72
16384	14	7.608	139,3
65536	16	8	272

Porównanie wartości funkcji logarytmicznej $\log_2 x$ oraz funkcji zapisanej wzorem /20/ wykazuje, że dla zbiorów mających 256 zapisów średnia liczba dostępów w razie zastosowania przeszukiwania interpolacyjno-skokowego jest mniejsza o 25% niż w przypadku przeszukiwania binarnego, a dla zbiorów o 65 536 zapisach wskaźnik ten wynosi 50%. Jednocześnie, mimo bardzo znacznej redukcji liczby dostępów dla najbardziej niekorzystnych rozkładów wartości identyfikatora, pozostaje ona duża w stosunku do wartości osiąganych przy przeszukiwaniu binarnym. Dlatego konieczne jest ścisłe określenie charakterystyk rozkładów niejednostajnych decydujących o średniej liczbie dostępów.

Podsumowanie

Opisane algorytmy, wykorzystujące ideę interpolacji do wyszukiwania zapisów w zbiorach i mających strukturę wektora, są mało znane i słabo wykorzystywane w istniejącym oprogramowaniu systemów komputerowych. Wynika to z kilku przyczyn, spośród których należy wymienić przede wszystkim dwie: brak szczegółowych opisów tych algorytmów w literaturze fachowej oraz słabo zbadana efektywność wyszukiwania dla niejednostajnych rozkładów wartości identyfikatora. Ten drugi czynnik uniemożliwia porównywanie interpolacji z innymi algorytmami, a w szczególności z przeszukiwaniem binarnym, które jest podstawową metodą wyszukiwania elementów w uporządkowanej tablicy jednowymiarowej.

Celem niniejszej publikacji było zaprezentowanie szczegółowych opisów algorytmów oraz zapoznanie z dotychczasowymi wynikami badań oceny ich efektywności[12]. Kolejnym etapem prac badawczych powinna być próba oceny efektywności opisanych algorytmów dla różnych rozkładów wartości identyfikatora w zbiorze.

[12] W pracy [14] autorzy ograniczyli się do bardzo ogólnego scharakteryzowania idei algorytmu. W niniejszym zaś artykule po raz pierwszy dokładnie opisano algorytm INTPRW.

LITERATURA

[1] S.P. G h o s h, M.E. S e n k o, File organization: On the selection of random accesses index points for sequential files, [w:] "Journal of the ACM", 1969, nr 1, s. 569-579.

[2] V.L. G o n č a r o v, Teorija interpolirovanija i priblizenija funkcij, Moskva 1954, Gos. Izd. Techn.-Teoret.Lit.

[3] G.H. G o n n e t, Interpolation and Interpolation Hash Searching. PhD thesis. University of Waterloo, Research Reports CS77-02 /January 1977/.

[4] G.H. G o n n e t, L.D. R o g e r s, The Interpolation - sequential search algorithm, "Informatin Processing Letters", 1977, nr 4, s. 136-139.

[5] G.H. G o n n e t, L.D. R o g e r s, J.A. G e o r g e, An Algorithmic and Complexity Analysis of Interpolation Search, "Acta Informatica", 1980, nr 1, s. 39-52.

[6] R.S. G u t e r, B.W. O w c z y ń s k i, Matematyczne opracowywanie wyników doświadczeń, Warszawa 1967, PWN.

[7] R.S. G u t e r, B.W. O w c z y ń s k i, Wstęp do metod numerycznych, Warszawa 1965, PWN.

[8] I.I. I b r a g i m o v, Metody interpolacii funkcij i nekatorye ich primenenija, Moskva 1971, Izd. Nauka.

[9] D.E. K n u t h, The Art of Computer Programming, vol 2, Seminumerical Algorithms. Reading, Massachusetts 1971.

[10] D.E. K n u t h, The Art of Computer Programming, vol 3 Sorting and Searching. Reading, Massachusetts 1973.

[11] J. K o r c z a k, Problemy budowy optymalnych struktur danych w pamięci dyskowej, Akademia Ekonomiczna, Wrocław 1978.

[12] J. K o r c z a k, Projektowanie zbiorów dyskowych, Akademia Ekonomiczna, Wrocław 1979,

[13] H.S. K r u i j e r, The Interpolated File Search Method, "Informatie", 1974, nr 7, s. 612-615.

[14] Y. P e r l, E.M. R e i n g o l d, Understanding the complexity of interpolation search,"Information Processing Letters", 1978, nr 7, s. 550-553.

[15] Y. P e r l, A. I t a i, H. A v n i, Interpolation Search - A LogLogN Search, "Comm. of ACM", 1978, nr 7.

[16] W.W. P e t e r s o n, Addressing for random - access storage, "IBM Journal of Res. and Develop", 1957 nr 1, s. 131--132.

[17] C.E. P r i c e, Table look - up techniques, "Computing Surveys", 1971, nr 2, s. 56-58.

[18] A. S t r z a ł k o w s k i, A. S l i ż y ń s k i, Matematyczne metody opracowywania wyników pomiarów, Warszawa 1978.

[19] S.J. W a t e r s, Analysis of self indexing disc files, "The Computer Journal", 1975, nr 1, s. 12-17.

[20] A.C. Y a o, F.F. Y a o, The complexity of searching an ordered random table, [w:] Proc. of 17 Annual IEEE Symposium on Foundations of Computer Science, Houston, 1976, s. 173-177.

[21] S. Z i m n o c h o, Wpływ blokowania na efektywność algorytmów wyszukiwania, w: Informatyka, Prace Naukowe Akademii Ekonomicznej we Wrocławiu /w druku/.

[22] S. Z i m n o c h o, Wyszukiwanie interpolacyjne w sieciowych bazach danych, /maszynopis na prawach rękopisu/.

APPLICATION OF INTERPOLATION FOR SEARCHING IN A SET OF RECORDS

Summary

The article presents an interpolative algorithm for identifier searching in a set of records with an orderly one-dimensional table structure. Two modifications of this algorithm are also described in detail: interpolative-sequential and interpolative-stepwise. Formulae enabling one to estimate the average number of accesses to an identifier's set of the uniform distribution are also given for each of the mentioned modifications.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Adam P a p s t
Akademia Ekonomiczna we Wrocławiu

ZASADY DOBORU TECHNIKI ZBIERANIA DANYCH DLA WSADOWYCH SYSTEMÓW INFORMATYCZNYCH

Problem doboru techniki zbierania danych odgrywa tym większą rolę, im większe wymagania stawiane są przed systemem informatycznym i im bardziej jest rozbudowany zestaw urządzeń technicznych do zbierania danych dostępny przyszłemu użytkownikowi systemu informatycznego. O specyfice problemu doboru techniki zbierania decyduje zbieranie danych w systemie informacyjnym obiektu, które spełnia niejako rolę łącza spajającego źródła danych z systemem komputerowym lub innym urządzeniem przetwarzającym dane.

Na proces doboru techniki zbierania danych wpływa wiele różnorodnych czynników. W artykule częściowo wykorzystano charakterystykę tych czynników przedstawioną przez L. Heinricha [2, s. 136 i n.] i Ch. Mellera [4, s. 95 i n.]. Czynniki wpływające na dobór techniki zbierania danych podzielimy na trzy grupy:

- niezależne od specyfiki obiektu i specyfiki systemu informatycznego,
- wynikające ze specyfiki systemu informatycznego,
- wynikające ze specyfiki obiektu.

Charakterystykę czynników wpływających na dobór techniki zbierania danych dla systemów informatycznych rozpoczniemy od czynników niezależnych od specyfiki obiektu i od specyfiki systemu informatycznego. Ta grupa czynników jest stosunkowo nieliczna, ale jej oddziaływanie jest dość znaczne. Za najważniejsze czynniki należące do tej grupy uznaje się:

- maksymalne koszty, jakie można ponieść na zakup i eksploatację urządzeń do zbierania danych,

- najdłuższy dopuszczalny okres wdrażania techniki zbierania danych,
- wymagany stopień adaptacji techniki zbierania danych do zmieniających się warunków systemu informacyjnego obiektu,
- wymaganą możliwość rozwoju systemu zbierania danych w miarę wzrostu wolumenu zbierania danych,
- przewidywany okres żywotności systemu zbierania danych,
- dostępny, w sensie możliwości zakupienia przez użytkownika systemu informatycznego, zestaw urządzeń do zbierania danych.

Czynniki zawarte w tej grupie wynikają z uwarunkowań zewnętrznych oraz z założeń strategicznych dotyczących rozwoju systemu informatycznego obiektu i rozwoju samego obiektu.

Następną grupę czynników wpływających na dobór techniki zbierania danych stanowią czynniki wynikające ze specyfiki systemu informatycznego. Proponujemy zaliczyć do tej grupy:

- największy dopuszczalny okres między powstaniem danych a ich przetworzeniem i przekazaniem wyników przetwarzania użytkownikowi systemu informatycznego,
- wynikający z poprzedniego czynnika sposób przetwarzania /wsadowy albo natychmiastowy/,
- wolumen danych podlegających zbieraniu, a tym samym i przetwarzaniu,
- zestaw urządzeń wejściowych komputera wykorzystywanego do eksploatacji systemu informatycznego,
- dopuszczalne ze względu na zestaw urządzeń wejściowych maszynowe nośniki danych,
- kody do zbierania danych akceptowane przez urządzenia wejściowe komputera przetwarzającego,
- wynikające z ustaleń projektowych długości i struktury zapisów danych podlegających zbieraniu.

Czynniki, które wpływają na dobór techniki zbierania danych i wynikają ze specyfiki systemu informatycznego są pochodną wymagań systemu informacyjnego obiektu i systemu zarządzania obiektem. Inne czynniki wiążą się wyłącznie ze sprzętem.

Ostatnią grupę czynników wpływających na dobór techniki zbierania danych do systemu informatycznego tworzą czynniki wynikające ze specyfiki obiektu. Naszym zdaniem obejmuje ona między innymi:

- rozproszenie źródeł danych,
- mobilność źródeł danych /stacjonarne lub ruchome/,
- natężenie spływu danych w poszczególnych źródłach danych,
- wyposażenie źródeł danych w urządzenia techniczne,
- warunki zewnętrzne źródeł danych /hałas, temperatura, wilgotność, zabrudzenie, wykorzystywana przestrzeń itp./,
- dostępność źródła danych dla człowieka,
- postać powstających danych /cyfrowe albo analogowe/,
- rodzaj zbieranych danych /numeryczne, alfabetyczne, alfanumeryczne/,
- sposób ustalania i rejestrowania danych,
- kwalifikacje personelu ustalającego i rejestrującego dane w źródłach,
- czas, który ten personel może przeznaczyć na zbieranie danych bez ujemnego wpływu na wykonywanie swoich zasadniczych obowiązków,
- obowiązujące w danym obiekcie przepisy prawne, finansowe, księgowe, statystyczne itp.

Zatem czynniki wynikające ze specyfiki obiektu dotyczą głównie źródeł danych, samych danych, personelu oraz różnego rodzaju przepisów obowiązujących w obiekcie.

Proponujemy następujące podejście do problemu doboru techniki zbierania danych. Najpierw użytkownik powinien przeanalizować wszystkie czynniki w celu ich policzenia lub opisania. Następnie należy te czynniki uporządkować w kolejności wynikającej z rangi przyznawanej im przez użytkownika systemu informatycznego. Efektem analizy tych czynników oraz ich uporządkowania będzie indywidualna lista ograniczeń, która powinna być podstawą do wyboru techniki zbierania danych. Użytkownik porównuje ograniczenia zawarte w liście z odpowiednimi charakterystykami technik zbierania danych. Należy podkreślić, że proces ten ma zwykle charakter iteracyjny. Może się bowiem okazać, że żadna z technik nie spełnia wszystkich wymagań użytkownika systemu informatycznego. Wówczas kontynuuje się poszukiwania, rezygnuje z kolejnych, mniej istotnych wymagań użytkownika. Gdy ostatecznie kilka dobranych technik w jednakowym stopniu spełnia określone wymagania użytkownika systemu informatycznego, wówczas należy wybrać tę, która zapewnia najmniejsze jednostkowe koszty zbierania danych.

W literaturze przedmiotu rzadko podejmowano problem doboru technik zbierania danych. Bardzo interesującą próbę rozwiązania tego zagadnienia przedstawił L. Heinrich w pracach 2,3 . Proponuje on model teoretyczny, który naszym zdaniem - jest mało przydatny dla użytkownika systemu informatycznego. Sądzimy, że proponowane przez nas podejście ma znacznie większe szanse na wdrożenie.

W dalszym ciągu naszych rozważań zajmiemy się procesem doboru techniki zbierania danych dla wsadowych systemów informatycznych. Oczywiście przedstawione ogólne zasady doboru odnoszą się także do systemów wsadowych, z tym, że czynniki wpływające należy porządkować według określonej hierarchii. Ponadto zestaw technik zbierania dla systemów wsadowych stanowi tylko pewną część dostępnych technik zbierania danych. Ma to wpływ na proces doboru.

Dokonamy obecnie analizy czynników wpływających na proces doboru techniki zbierania danych z punktu widzenia wsadowych systemów informatycznych. Uwzględnimy specyficzne warunki krajowych użytkowników tych systemów.

Rozważania rozpoczniemy od analizy czynników niezależnych od specyfiki zarówno obiektu, jak i systemu informatycznego.

Ograniczenia dotyczące kosztów zakupu i eksploatacji technik zbierania danych dla systemów wsadowych odgrywają mniejszą rolę niż przy technikach zbierania danych dla systemów natychmiastowych, gdyż koszty zbierania są znacznie mniejsze. W kraju natomiast ze względu na trudności dewizowe użytkownik systemu informatycznego nie może nabyć określonych urządzeń do zbierania danych. Ponadto pewnym utrudnieniem przy doborze technik zbierania danych jest brak informacji o kosztach zbierania dla różnych technik. Użytkownik systemu informatycznego dobierający techniki zbierania musi się, niestety, kierować subiektywnym wyczuciem w sprawie wielkości kosztów zbierania danych. Tylko w przypadku korzystania z obcych usług w zakresie zbierania danych użytkownik może posłużyć się obowiązującymi cennikami usług informatycznych do oszacowania kosztów zbierania danych.

Kolejny czynnik tej grupy to maksymalny okres wdrażania techniki zbierania danych. Czynnik ten dla technik zbierania wykorzystujących nośniki maszynowe ma różne znaczenie. Zależy

ono od związku między czynnością rejestrowania i czynnością zbierania. Jeśli obie czynności wykonywane są automatycznie, to okres wdrażania jest krótki. Jeśli zaś czynności te wykonuje człowiek, to okres wdrażania jest dłuższy. W porównaniu z technikami zbierania danych dla systemów natychmiastowych techniki zbierania stosujące nośniki maszynowe zapewniają krótszy okres wdrażania.

Następne dwa czynniki: stopień adaptacji techniki zbierania danych do zmieniających się warunków systemu informacyjnego obiektu oraz możliwość rozwoju systemu zbierania danych w miarę wzrostu wolumenu zbieranych danych można potraktować łącznie i określić mianem elastyczności techniki zbierania danych. Uważamy, że techniki zbierania danych dla systemów wsadowych są bardziej elastyczne niż techniki zbierania danych dla systemów natychmiastowych, gdyż mają większe możliwości rozbudowy swych konfiguracji ze względu na fizyczne odizolowanie urządzeń zbierających od przetwarzającego systemu komputerowego.

Przewidywany okres żywotności systemu zbierania danych wynika z założeń strategicznych dotyczących rozwoju systemu informatycznego i zwykle uważa się go za czynnik o niewielkim znaczeniu dla procesu doboru.

Dla użytkownika systemu informatycznego bardzo ważnym czynnikiem jest zestaw urządzeń do zbierania danych, gdyż decyduje on o czasochłonności procesu doboru. Niewielki zestaw urządzeń zbierających dostępnych dla krajowego użytkownika znacznie skraca czas omawianego procesu.

Charakterystykę czynników wynikających ze specyfiki systemu informatycznego rozpoczniemy od czasu, jaki może upłynąć od powstania danych do ich przetworzenia. Czynnik ten odgrywa znacznie mniejszą rolę w technikach zbierania dla systemów wsadowych niż dla systemów natychmiastowych. Czas między powstaniem danych a ich przetworzeniem znajduje odzwierciedlenie w harmonogramie przetwarzania systemu informatycznego. Różny czas w poszczególnych technikach zbierania dla systemów wsadowych wynika ze stopnia decentralizacji zbierania danych, funkcjonalnych możliwości urządzeń oraz stopnia integracji zbierania danych z poprzedzającymi go czynnościami.

O wyborze techniki zbierania danych decydują wolumen danych podlegających zbieraniu oraz czas, który personel ustalający i rejestrujący dane w ich źródłach może przeznaczyć na te czynności. Jeśli wolumen spływających danych będzie tak duży, że uniemożliwi zbieranie danych u ich źródła, to w procesie doboru zostaną wyeliminowane techniki zakładające zbieranie danych w miejscu ich powstawania. Podobnie będzie wówczas, gdy wolumen spływających danych będzie niewielki.

Zestaw urządzeń wejściowych komputera przetwarzającego określa dopuszczalne maszynowe nośniki danych oraz stosujące je urządzenia zbierające.

Ostatni czynnik z tej grupy to długość i struktury zapisów danych podlegających zbieraniu. Mają one znaczenie zwłaszcza dla technik zbierania stosując nośniki pojedyncze, czyli dla takich, w których każdy zapis danych zapamiętywany jest na fizycznie samodzielnym nośniku. Wtedy pojemność informacyjna nośnika pojedynczego określa maksymalne długości zapisów. Ponadto ograniczenia długości i struktury zapisów występują w przenośnych urządzeniach dla zbierania danych [1, s. 8; 6, s. 59].

Rozważania nad analizą czynników wpływających na proces doboru techniki zbierania danych dla wsadowych systemów informatycznych zakończymy charakterystyką czynników wynikających ze specyfiki obiektu. Pierwszy z czynników tej grupy to rozproszenie źródeł danych, które decydują o możliwości zainstalowania urządzeń zbierających dane u źródeł. Czynnik ten łączy się ściśle z omówionym wcześniej natężeniem spływu danych w poszczególnych źródłach.

Mobilność źródeł danych decyduje o ewentualnym zainstalowaniu urządzeń zbierających dane. Gdy źródło danych jest ruchome, to wybór spośród urządzeń zbierających jest bardzo ograniczony.

Wyposażenie źródeł danych w środki techniczne służące do ustalenia i rejestrowania danych oraz sposób ustalania i rejestrowania danych decydują o sposobie zbierania oraz możliwości zintegrowania zbierania z poprzedzającymi go czynnościami. Tym samym wpływają na wybór techniki zbierania danych.

Na temat warunków zewnętrznych źródeł danych można powiedzieć, że im bardziej są one niekorzystne /wysoka temperatura,

duży hałas, duża wilgotność itp./, to tym prostsze w konstrukcji i obsłudze muszą być urządzenia do zbierania danych.

Niedostępność źródła danych dla człowieka wymusza powierzenie zbierania urządzeniom technicznym, czyli wymusza automatyzację zbierania danych.

Postać powstających danych narzuca stosowanie urządzeń techniki cyfrowej lub techniki analogowej. W technice cyfrowej rodzaj zbieranych danych sprawia, że wybiera się urządzenia zbierające o określonym repertuarze znaków.

Zarówno poziom kwalifikacji personelu ustalającego i rejestrującego dane oraz czas, który ten personel może przeznaczyć na realizację czynności zbierania, określa dopuszczalny stopień skomplikowania obsługi urządzeń zbierających.

Ostatni z wyróżnionych czynników tej grupy to obowiązujące w danym obiekcie przepisy prawne, finansowe, księgowe, statystyczne itp. Znaczenie ich polega na tym, że to właśnie one wymagają zarejestrowania zachodzących zjawisk gospodarczych w postaci dokumentów źródłowych. Dokumenty te muszą spełniać wymagania narzucone przez wymienione przepisy. Z tego względu należy wybrać taką technikę zbierania danych, która opierałaby się na dokumentach źródłowych spełniających powyższe wymagania.

Dotychczasowe rozważania koncentrowały się na kierunkach oddziaływania poszczególnych czynników wpływających. Zmierzano bowiem do ukazania roli każdego z tych czynników przy doborze techniki zbierania danych dla wsadowych systemów informatycznych.

W wyniku analizy czynników wpływających użytkownik uzyskuje indywidualny zestaw tych ograniczeń. W następnym kroku porównuje swój zestaw ograniczeń z odpowiednimi parametrami dostępnych urządzeń zbierających oraz analizuje koszty zbierania danych. Gdy jednak uzyska kilka równorzędnych technik zbierania, to o wyborze powinien zadecydować poziom usług producenta urządzeń zbierających. Do usług tych proponujemy zaliczyć:

- usuwanie awarii,
- konserwację urządzeń,
- tworzenie oprogramowania podstawowego i użytkowego /w przypadku programowanych urządzeń zbierających/,
- szkolenie personelu,
- organizację eksploatacji.

Na zakończenie rozważań wymienimy parametry techniki zbierania danych, które - naszym zdaniem - mają podstawowe znaczenie dla procesu doboru techniki zbierania danych dla wsadowych systemów informatycznych. Oto one:

- automatyczne zbieranie,
- połączenie czynności zbierania z czynnością ustalania i rejestrowania danych,
- instalowanie urządzenia zbierającego w miejscu powstawania danych /wymagane warunki zewnętrzne i natężenie spływu danych itp./,
- wymagany poziom kwalifikacji personelu zbierającego dane,
- właściwości stosowanego nośnika danych,
- stopień inteligencji urządzenia zbierającego.

Z przedstawionej w artykule koncepcji realizacji procesu doboru techniki zbierania danych można korzystać wówczas, gdy rynek jest bogato zaopatrzony w urządzenia do zbierania danych.

LITERATURA

[1] W. de B e a c l a i r, Tragbare Geräte zur mobilen Datenerfassung, "Bürotechnische Sammlung", 1977, nr 1.

[2] L.J. H e i n r i c h, Planung des Datenerfassungssystems. Bd.2. der Schriftenreihe "Methoden der Planung und Lenkung von Informationssystemen". Köln 1974, Verlagsgesellschaft Rudolf Müller.

[3] L.J. H e i n r i c h, Planung und Optimierung des Datenerfassungsystems. Tübingen 1980, Integrata GmbH.

[4] Ch. M e l l e r, Grundlagen und organisatorische Möglichkeiten der Datenerfassung, Wiesbaden 1972, Verlag Dr.Th. Gabler.

[5] A. P a p s t, Pewne aspekty klasyfikacji i wyboru systemów zbierania danych.[w:] Prace Naukowe Akademii Ekonomicznej we Wrocławiu,nr 98. Wrocław 1977, AE.

[6] A. P a p s t, Przenośne urządzenia do zbierania danych, "Informacje i Komunikaty ZETO Wrocław", 1978, nr 1.

THE PRINCIPLES OF THE SELECTION OF A DATA COLLECTING TECHNIQUE FOR BATCH COMPUTER SYSTEMS

Summary

The paper contains a concept for making the selection of a data collecting technique for batch computer systems. Three groups of factors exerting an influence on making this selection are presented and discussed:

1. factors independent of the specificity of the object and the specificity of the computer system,
2. factors resulting from the specificity of the computer system, and
3. factors resulting from the specificity of the object.

Then a general algorithm is presented for making of the selection. This algorithm is applied to the discussion of the selection of a data collecting technique for batch computer systems.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Barbara Ł u k a s i k - M a k o w s k a
Akademia Ekonomiczna we Wrocławiu

MOŻLIWOŚCI ZASTOSOWANIA KOMPUTERA DO PRAC ZWIĄZANYCH Z PROJEKTOWANIEM ODZIEŻY W PRZEMYŚLE

1. Problemy zaopatrzenia w odzież

Z uwagi na warunki klimatyczne, w jakich żyjemy, odzież jest nam niezbędna do ochrony przed zimnem, wiatrem i opadami, a czasem nawet i słońcem. Dodatkowo tradycje kulturowe określają minimum odzieży, w jaką należy być ubranym, mimo iż w sprzyjających okolicznościach odzież może nie być niezbędna do ochrony przed niesprzyjającymi czynnikami meteorologicznymi. Tradycje kulturowe odmienne na różnych szerokościach geograficznych ulegają dodatkowo modzie, która w dość krótkim okresie modyfikuje zarówno kategorie odzieży /np. nosi się spodnie a nie nosi spódnic itp./, jak jej kolorystykę czy sposób wykonania. Ponadto sylwetka każdego człowieka ulega zmianom: we wczesnych okresach życia zwiększa się wzrost i objętość ciała, następnie zmieniają się proporcje między poszczególnymi obwodami, potem zwiększają się obwody i zniekształca sylwetka na skutek niewłaściwego trybu życia.

Tak więc każdy człowiek potrzebuje co pewien czas nowej odzieży na skutek zużycia dotychczas posiadanej z powodu zmiany pory roku, zmiany sylwetki lub też chęci nadążenia za modą. Oczywiście dominacja poszczególnych czynników będzie różna u różnych osób w zależności od płci, wieku i możliwości finansowych, a często również od charakteru pracy lub miejsca zamieszkania. I tak, w grupie wiekowej dzieci i młodzieży najczęstszymi przyczynami wymiany odzieży jest zmiana sylwetki oraz zużycie odzieży. Należy zauważyć, że wpływ tych czynników na różne

części odzieży jest nierównomierny /np. chłopcy najszybciej niszczą spodnie/.

Z punktu widzenia kobiet jednym z silniejszych czynników wpływających na zmianę odzieży jest zmieniająca się moda. Mężczyźni natomiast kierują się zwykle praktycznością i zazwyczaj wyrażają zapotrzebowanie na nową odzież w związku z nowymi okolicznościami życiowymi lub zawodowymi.

Do wyrobu odzieży używa się tkanin z włókien naturalnych i syntetycznych. Niektóre mogą się nadawać na przykład do wykonania tylko jednego rodzaju garderoby /np. tzw. tkaniny płaszczowe/, inne - uniwersalne - mogą służyć do produkcji różnego rodzaju odzieży.

Rodzaj tkaniny zazwyczaj decyduje o przeznaczeniu konkretnej części odzieży na określoną porę roku /np. w przypadku płaszcza/, może też wpłynąć na uniwersalność odzieży i umożliwiać użytkowanie jej w ciągu całego roku /np. koszule męskie z elanobawełny/.

Wszystkie wymienione czynniki pozwalają stwierdzić, że ludzie stale potrzebują nowej odzieży dopasowanej do ich figur, dającej odpowiednie zabezpieczenie przed warunkami atmosferycznymi oraz odpowiadającej indywidualnym gustom, upodobaniom, a także modzie.

2. Etapy powstawania odzieży

Wszystkie wymienione czynniki sprawiają, że konieczne jest produkowanie bogatego i zmieniającego się asortymentu odzieży, zaspokajającego wymienione potrzeby.

Czy jednak są one zaspokajane? Posiadaną przez nas odzież w zależności od miejsca i sposobu powstawania można podzielić na:

1/ produkowaną seryjnie przez przemysł odzieżowy,

2/ produkowaną indywidualnie przez uspołecznione punkty usługowe,

3/ szytą indywidualnie przez prywatny zakład krawiecki,

4/ szytą samemu bądź przez kogoś z rodziny.

W poszczególnych grupach wieku i płci, możliwości finansowych oraz rodzaju odzieży dominuje jeden z wymienionych sposobów pokrycia zapotrzebowania na odzież. Przedstawiony podział ma ścisły związek z przebiegiem procesu powstawania odzieży.

Etapy powstawania odzieży różnią się istotnie w zależności od tego, czy odzież stanowi indywidualny egzemplarz przeznaczony dla konkretnej osoby i w związku z tym jest dopasowana do cech figury tej osoby czy też stanowi odzież "anonimową" przeznaczoną dla osób o wymiarach podstawowych /wzrost i określone obwody, np. pasa, klatki piersiowej, bioder/ zgodnych z określonym wzorcem.

W produkcji przemysłowej mamy oczywiście do czynienia wyłącznie z sytuacją drugą[1].

W procesie powstawania odzieży można wyróżnić następujące etapy:

1. Opracowanie plastycznego projektu elementu odzieży.
2. Przygotowanie szablonów wykrojów.
3. Przygotowanie rozkładu szablonów na tkaninie.
4. Rozkrój tkaniny.
5. Szycie i wykańczanie odzieży.

Etap pierwszy realizują zazwyczaj specjaliści plastycy - projektanci odzieży. Efekt ich pracy odzwierciedla zarówno modę, twórczą fantazję projektanta, jak i potrzeby praktyczne w odniesieniu do danej części odzieży. Wyspecjalizowana komisja zatwierdzająca projekty do przemysłowej produkcji odzieży uwzględnia takie cechy modelu, jak:

- efekt plastyczny,
- przydatność praktyczna,
- łatwość produkcji seryjnej,
- możliwość zakupu odpowiednich materiałów, dodatków itp.

Dla zaakceptowanych modeli opracowuje się szablon, czyli dokonuje się "rozłożenia" modelu na elementy składowe, które muszą być wykonane z materiału podstawowego, dodatkowego lub innych elementów uzupełniających /np. zamków/. Szablon jest wykonany dla jednego, średniego rozmiaru dla określonej grupy ludzi przewidywanych jako użytkownicy danego elementu odzieży.

Ponieważ produkuje się dla większej liczby rozmiarów, konieczne jest przeprowadzenie tzw. stopniowania szablonów, czyli opracowania szablonów dla wszystkich rozmiarów, które mają być produkowane.

[1] Tabele wymiarów dla obu płci i różnych grup wiekowych opracowane są na podstawie badań antropometrycznych i zawierają modelowe dane dla różnych wzrostów i obwodów oraz typów sylwetki A, B, C i D. Por. [1, s. 42, 69].

Kolejnym, trzecim etapem jest opracowanie rozkładu szablonów na tkaninie. Ma ono zapewnić minimalne zużycie materiału na jednostkę odzieży. Istotne jest tu takie wykorzystanie elementów szablonów różnych rozmiarów, aby z określonej ilości tkaniny można było uzyskać jak największą liczbę kompletnych elementów odzieży. Planowanie rozkładu szablonów na tkaninie pozwala na określenie norm zużycia materiału. Należy jednak zaznaczyć, że w czasie produkcji na skutek niejednolitej szerokości tkaniny w stosunku do projektowanej /wystarcza różnica ± 2 cm/ rozkład musi być modyfikowany, aby można było dotrzymać zaplanowanych norm. Chociaż w teorii znane jest zagadnienie optymalizacji rozkroju, w praktyce przemysłu odzieżowego jest ono rozwiązywane wyłącznie intuicyjnie [3].

Na podstawie wybranego optymalnego rozkładu szablonów dokonuje się rozkroju tkaniny. Precyzyjne wykonanie tej czynności zapewnia zgodność poszczególnych elementów i prawidłowe uszycie części odzieży.

Ostatnim etapem jest wykonywanie odzieży, czyli połączenie wszystkich elementów według ustalonej kolejności, wykonanie określonych prac wykończeniowych /np. przyszywanie guzików, obszywanie i przecinanie dziurek/, uprasowanie i zapakowanie gotowej części do wysyłki.

Przedstawiony proces powstawania odzieży jest w praktyce nieco bardziej skomplikowany ze względu na istnienie pewnych funkcji wspomagających bezpośredni proces produkcyjny /obsługa techniczna maszyn, kontrola techniczna, służby księgowe, gospodarka magazynowa itp./. W dalszych rozważaniach nie będziemy tych funkcji uwzględniać, gdyż mogą one być adaptowane do wymagań i przebiegu głównego procesu produkcyjnego.

3. Możliwości zastosowania komputera w przemyśle odzieżowym

Zapotrzebowanie społeczne na nową i dostosowaną do indywidualnych wymagań odzież jest zatem w pewnej pozornej sprzeczności z interesem przemysłu odzieżowego, dla którego najwygodniejsze jest produkowanie długich serii odzieży o stosunkowo małym asortymencie modeli i niewielkim zróżnicowaniu rozmiarów. Gdyby jednak przemysł produkował znikomą liczbę fasonów i rozmiarów odzieży, to sklepy i magazyny byłyby zapełnione odzieżą, na którą

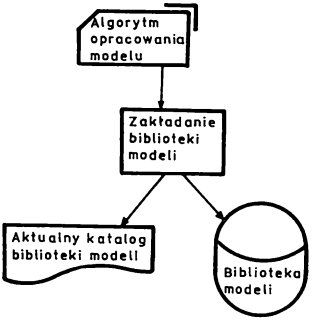
trudno byłoby znaleźć nabywców. Kierujący przemysłem wiedzą jednak, że krótkie serie i częsta zmiana modeli /dodatkowo zgodnych z zapotrzebowaniem społecznym/ mają istotny wpływ na popyt, którym będzie się cieszyć wyprodukowana odzież. Pewne mechanizmy ekonomiczne, którymi do chwili obecnej rządzony jest przemysł i handel, powodują jednak, że popyt nie zyskał w przemyśle odzieżowym dostatecznie wysokiej rangi, czego odzwierciedleniem jest fakt, że na półkach i wieszakach znajduje się odzież, która nie może znaleźć nabywców, a jednocześnie niezadowoleni klienci płci obojga, w różnym wieku, mają bardzo istotne trudności ze zdobyciem dla siebie odpowiedniej garderoby.

Wydaje się, że jedną z istotnych, a zarazem możliwych do przezwyciężenia przeszkód ograniczających elastyczność asortymentową produkcji, jest długotrwały, a zarazem drogi tryb przygotowania nowego modelu do produkcji. Proces tem można znacznie usprawnić i przyspieszyć, wykorzystując do jego realizacji /bądź co najmniej wspomagania/ system komputerowy.

Wykorzystanie komputera pozwoli bowiem na rozszerzenie liczby wykrojów dla różnych, możliwie zbliżonych do indywidualnych wymiarów, a także na zwiększenie gamy produkowanych fasonów. Kolejną korzyść może także stanowić usprawnienie procesu technologicznego przygotowania produkcji, a w tym tak pracochłonnych elementów, jak normowanie zużycia materiałów, planowanie produkcji i zaopatrzenia czy nawet niedoceniane dotychczas zagospodarowywanie resztek metrażowych. Wszystkie te funkcje mogą w przedsiębiorstwie wpłynąć na obniżenie jednostkowych kosztów produkcji oraz na zwiększenie zysków przedsiębiorstwa z racji lepszego wykorzystania materiału oraz produkowania wyrobów, na które oczekuje społeczeństwo.

Uzyskiwanie wykrojów modeli odzieży dla nieomal indywidualnych klientów jest możliwe ze względu na wykorzystanie znanych w literaturze zasad obliczania poszczególnych odcinków konstrukcyjnych wykroju [2]. Obliczeń tych dokonuje się za pomocą stosunkowo prostych algorytmów, do których jako dane wejściowe wykorzystuje się podstawowe wymiary konkretnej osoby zdjęte w określonych dla danej kategorii odzieży miejscach /np. obwód pasa, bioder, długość spódnicy, spodni itp./. Liczba tych wymiarów dla konkretnego elementu odzieży jest niewielka, od 3 do 10. Algorytmy obliczeniowe można zatem zgrupować w bibliotece pro-

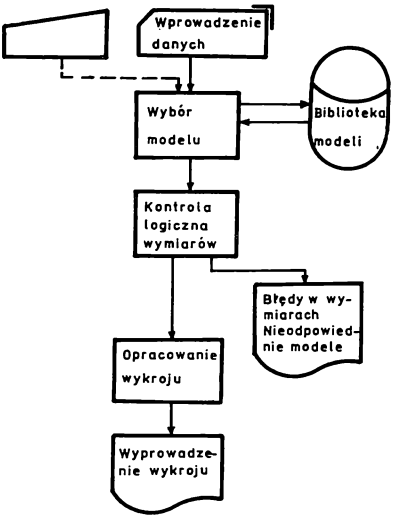
gramów i identyfikować za pomocą symbolu modelu /rys. 1/. Wykonanie wykroju powinno być poprzedzone przygotowaniem i wprowadzaniem zaledwie 1 karty dziurkowanej z danymi /w przypadku modeli bardziej złożonych 2 kart/.



Rys. 1. Zakładanie i aktualizacja biblioteki modeli

System taki będzie oczywiście tym bardziej użyteczny, im więcej modeli /algorytmów/ zgromadzi się w bibliotece. Zakłada się, że bibliotekę można systematycznie powiększać przez uzupełnianie nowymi algorytmami uzyskiwania wykrojów. Bibliotekę można przechowywać zarówno na dysku, jak i taśmie magnetycznej. Zależy to od częstotliwości jej wykorzystania.

Każdy z algorytmów opracowania wykroju składa się z dwu części /rys. 2/. Jedną jest sprawdzenie logicznej poprawności wprowadzonych wymiarów /wynikającej z istniejących proporcji ciała/ oraz badanie, czy określony model może być opracowany dla tych wymiarów /istnieją bowiem pewne anomalie budowy ciała, które powodują, że określony model wyglądałby na takiej figurze bardzo niekorzystnie/. Drugą część stanowi przygotowanie i wyrysowanie wykroju.



Rys. 2. Schemat komputerowego opracowywania wykrojów

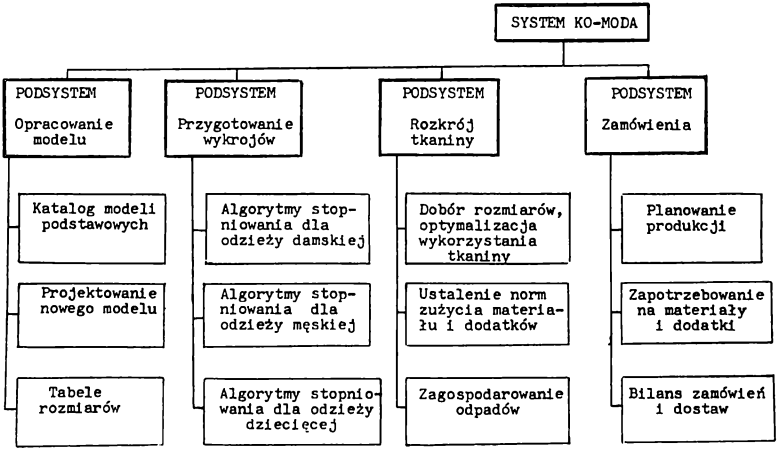
Należy zauważyć, że kształtowanie fasonu odzieży to nie tylko jego konstrukcja dająca podstawowy kształt garderoby, ale również modelowanie, którego zadaniem jest opracowanie wszystkich elementów zdobniczych, jak kieszenie, patki, cięcia, stebnowania itp. A zatem z jednego modelu podstawowego możemy uzyskać wiele wersji różnie wymodelowanej określonej części odzieży. Modelowanie takie może odbywać się również za pomocą komputera. Korzystne byłoby realizowanie takich prac za pomocą systemu konwersacyjnego i przy użyciu grafoskopu. Możliwe jest jednak wprowadzenie danych o sposobie i zakresie modelowania jako danych wsadowych. Dotyczy to usytuowania kieszeni i patek, wprowadzenia karczków, cięć pionowych i poziomowych itp.

Inną możliwość zindywidualizowania wykrojów odzieży otrzymujemy przez wykorzystanie tzw. stopniowania szablonów. Termin ten powszechnie używany w przemyśle odzieżowym oznacza, że kolejne wielkości modeli konkretnej części garderoby nie są opracowywane zgodnie z wymiarami podstawowymi i przy użyciu odpowiednich algorytmów, lecz że tak opracowuje się jeden /wybrany/ rozmiar, a pozostałe, większe i mniejsze, uzyskuje się przez odpowiednie powiększanie bądź zmniejszanie wyjściowego szablonu. Technika ta pozwala na przyspieszenie opracowania potrzebnych szablonów, niesie jednak z sobą niebezpieczeństwo pewnych zniekształceń modelu, które mogą niekorzystnie wpłynąć na fason odzieży. W razie posługiwania się techniką komputerową korzystniej jest opracować wykroje każdego modelu oddzielnie niż jeden powiększać lub zmniejszać.

Ze stopniowania szablonów można natomiast korzystać wówczas, gdy nie mamy algorytmu opracowania wykroju. Dotyczy to zwłaszcza modeli bardzo fantazyjnych opracowanych za pomocą indywidualnego modelowania na manekinie. Otrzymany w ten sposób wykrój dopasowany do konkretnych wymiarów może stanowić dane wyjściowe, które poddane przetwarzaniu za pomocą odpowiednich algorytmów stopniowania pozwolą uzyskać wykroje modelu o pożądanych wymiarach. Istotne trudności nastręcza w miarę precyzyjne wprowadzenie rysunku. Byłoby to proste w przypadku posiadania czytników rysunków /nie produkuje się ich w kraju/. Jeżeli jednak dysponuje się tylko urządzeniami wejścia z MNI, to konieczne jest przeniesienie wykroju na arkusze papieru posiadające dokładny raster bądź też zapisanie współrzędnych charakterystycznych punktów wykroju w skali milimetrowej oraz określenie funkcji linii łączących te punkty. Jest to łatwe w odniesieniu do prostych odcinków linii, kłopotliwe natomiast w odniesieniu do łuków i krzywizn.

Algorytmy stopniowania są specyficzne dla płci, rozmiarów i rodzaju odzieży. W związku z tym muszą stanowić odrębną bibliotekę programów, z której poszczególne programy będą w miarę potrzeby realizowane.

W przyszłości przewiduje się konwersacyjne wykorzystanie zasobów oraz rozszerzenie funkcji systemu o zagadnienia związane z:



Rys. 3. Koncepcja struktury systemu komputerowego opracowywania odzieży

- ułożeniem wykroju na tkaninie /z uwzględnieniem jej wzoru i optymalizacji zużycia materiału/,

- planowaniem rozkroju przy jednoczesnej produkcji różnych rozmiarów,

- ustaleniem zapotrzebowania na materiały uzupełniające i dodatki krawieckie,

- propozycjami wykorzystania odpadów powstających w czasie rozkroju tkaniny.

Na rys. 3 przedstawiono koncepcję systemu KO-MODA uwzględniającą wszystkie omówione problemy.

LITERATURA

[1] Z. L i p k e - S k r a w e k, Konstrukcja i modelowanie odzieży lekkiej, cz. II. Warszawa 1976, WSIP.

[2] Z. P a r a f i a n o w i c z, Podstawy modelowania odzieży, Warszawa 1979, WNT.

[3] B. P i a s e c k i, Optymalny rozkrój materiałów, Warszawa 1979, WNT.

POSSIBILITIES FOR USING COMPUTERS LARGE-SCALE DRESS DESIGNING

Summary

Possibilities for using computers in the preparation of large-scale clothes production are presented in the paper. Problems connected with developing and exploiting the library of algorithms to be used for making various types of clothing patterns are widely discussed here.

PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 275 Informatyka 1984

Bernard K u b i a k
Uniwersytet Gdański

O PEWNYCH METODACH ANALIZY SYSTEMÓW INFORMATYCZNYCH

1. Wstęp

Analiza systemów informatycznych wywodzi się z analizy systemów informacyjnych. Jako samodzielna dyscyplina analiza systemów informacyjnych wyodrębniła się dopiero w końcu lat pięćdziesiątych. Jednakże znaczny wzrost popytu na ten rodzaj wiedzy zaznaczył się wyraźnie w okresie gwałtownego rozwoju zastosowań komputerów w zarządzaniu. W miarę rozwoju komputeryzacji procesu decyzyjnego zaczął upowszechniać się pogląd, iż zarządzanie oraz procedury postępowania stosowane w przedsiębiorstwach przemysłowych i handlowych mogą być s t a n d a r d o w o u s p r a w n i a n e przez odpowiednio rozbudowaną i dobrze zorganizowaną a n a l i z ę ich funkcji. Wkrótce jednak okazało się, że znaczna część takich usprawnień wymaga w większym stopniu zastosowania "dyscypliny" niż "instrumentu" /za jaką początkowo uchodziła analiza systemów/. Skuteczne zastosowanie "dyscypliny" wymagało uprzedniego opracowania odpowiednich metod i technik analitycznych oraz n o w e j technologii przetwarzania danych. Zadośćuczynienie tym wymaganiom doprowadziło do powstania nowej dyscypliny, zwanej a n a l i z ą s y s t e m ó w i n f o r m a c y j n y c h, która stopniowo - ze względu na zawężanie przedmiotu swych zainteresowań - zaczęła się przeradzać w a n a l i z ę s y s t e m ó w i n f o r m a t y c z n y c h.

Dotychczasowe doświadczenia wskazują, iż analiza systemów informacyjnych rozwinęła się i upowszechniła w praktyce mimo dotkliwego braku literatury przedmiotu. Późniejszy rozwój teorii analizy systemów informacyjnych stale pozostawał w dyspro-

porcji z praktycznymi wskazówkami dla analitów systemów. Utrudniło to analitykom zdobycie odpowiedniego doświadczenia w badaniu systemów informacyjnych i systemów informatycznych - zwłaszcza umiejętności stosowania w praktyce m e t o d identyfikacji, badania i tworzenia systemów informatycznych /SI/, głównie zaś rejestrowania faktów i zdarzeń - i stało się czynnikiem nie sprzyjającym nawiązaniu przez twórców efektywnej współpracy z użytkownikami SI.

W niniejszym opracowaniu podjęto próbę krytycznego spojrzenia na te zagadnienia oraz wskazania sposobów poprawienia istniejącej sytuacji.

2. Kierunki analizy systemów informatycznych

Na podstawie przeglądu publikacji poświęconych analizie systemów oraz badań tych zagadnień nietrudno zauważyć, iż analiza systemów informatycznych /ASI/ jest u nas nadal nie doceniana. Dzieje się tak pomimo to, iż czas i koszty projektowania oraz koszty realizacji SI stale wzrastają. Nie pozostaje to bez wpływu na efektywność zastosowań systemów informatycznych zarządzania /SIZ/, a ściślej, na efektywne dostarczanie informacji poszczególnym szczeblom decyzyjnym /w odpowiednie miejsca/ z zastosowaniem właściwych środków i metod opracowywania danych, algorytmów i programów.

Realizacja tych postulatów stała się naczelnym zadaniem ASI, ponieważ prowadzi do zdefiniowania SI, określenia podstawowych zbiorów informacji w systemie oraz elementarnych procesów, które zapewnią uzyskiwanie dowolnego zestawu informacji. Wynika z tego, iż ASI jest n a r z ę d z i e m ich tworzenia i że powinna stać się obowiązkową fazą p r z e d p r o j e k t o w ą i zaczynać od rozpoznania poprzedzającego właściwe tworzenie SI. Tym sposobem staje się bowiem bardziej widoczne, "gdzie badać następnie" lub "co zrobić" i "jak to zrobić", aby usprawnić obieg informacji, natomiast analityk systemu żeby był dostatecznie przekonany o słuszności badania zarówno w aspekcie "dlaczego", jak też "co, kiedy, kto, gdzie i jak".

Zapewnienie stałego dopływu pożądanych informacji poszczególnym szczeblom decyzyjnym wymaga krytycznego podejścia do zasad tworzenia SI oraz stałego doskonalenia narzędzi ułatwiających ich projektowanie /por. też [5], s. 157 i n./. Wiąże

się to z koniecznością opracowywania n o w y c h z a s a d lub m o d y f i k o w a n i a s t a r y c h łącznie z narzędziami stosowanymi do realizacji SI.

Znawcy przedmiotu utrzymują, iż "narzędziowy" punkt widzenia analizy nie wystarcza do poprawnego tworzenia SI. Konieczne jest również zapewnienie dostatecznej spójności tego podejścia badawczego z:

1/ systemem zarządzania danego obiektu przez zapewnienie temu systemowi oczekiwanych informacji na odpowiednich stanowiskach i w odpowiednim czasie;

2/ modelowaniem matematycznym i ekonomicznym, a zwłaszcza z metodami analizy operacyjnej.

Ocenia się, że spójność wyrażona w punktach 1 i 2 n i e j e s t wystarczająca. Utrudnia to zaspokojenie potrzeb informacyjnych użytkowników SI. Częściowo sytuacja ta jest wynikiem stosowania zasad projektowania ISZ opracowanych w latach pięćdziesiątych. Nie trzeba być "odkrywcą", by stwierdzić, iż zasady te /znajdujące odzwierciedlenie w odpowiednich krokach i technikach projektowania/ n i e u w z g l ę d n i a j ą w dostatecznym stopniu postępu, jaki dokonał się w produkcji sprzętu obliczeniowego i jego oprogramowaniu. Chcąc uniknąć chociażby częściowo tych ograniczeń zaleca się stosowanie metody i t e r a c j i, a więc analizę informacji p o p r z e d z a j ą c y c h /wejściowych/ i informacji n a s t ę p u j ą c y c h /wyjściowych/. Istota tej metody polega na tym, że w każdym kroku po dokonaniu analizy systemu wyodrębnia się jego podsystemy, a następnie przeprowadza się analizę /lub projektowanie/ jednego z wybranych podsystemów. Takie podejście pozwala stwierdzić, czy informacja celowo i bezpośrednio wpływa na zachowanie użytkownika systemu, a zatem - czy jest c e l o w y m wyjściem informacyjnym systemu. Po tym badaniu łatwiej już stwierdzić, czy informacja tworzona w ramach ISZ jest informacją p o p r z e d z a j ą c ą dla obu wyżej rozpatrzonych przypadków.

Już na przykładzie analizy informacji poprzedzających i następujących są widoczne r e z e r w y zwiększenia skuteczności ISZ. Przy bliższym ich poznaniu okazuje się, że sprowadzają się one do prawidłowego grupowania:

a/ danych w zbiory lub "porcje",

b/ cząstkowych procesów w programy,

c/ informacji podstawowych w pozycje zagregowane.

W podsumowaniu dotychczasowych rozważań dotyczących kierunków ASI można stwierdzić, że powinny one doprowadzić do:

a/ powstania adaptacyjnych systemów informatycznych, którym projektanci nadaliby właściwości "uczenia się",

b/ powstania aktywnych systemów informatycznych,

c/ powstania systemów informatycznych nie tylko dostarczających użytkownikowi informacji o jego stanie, lecz także sterujących zachowaniem użytkownika podczas wykorzystywania tych stanów,

d/ powstania systemów informatycznych samodzielnie organizujących swoją realizację w zależności od sposobu wykorzystania i zachowania otoczenia,

e/ powstania systemów informatycznych przystosowujących się do warunków wynikających z zastosowania nowego sprzętu obliczeniowego,

f/ powstania technik i metod tworzenia SI o właściwościach wymienionych w punktach od a do c.

Uwzględniając te właściwości łatwo zauważyć, że chodzi o zaprojektowanie l e p s z y c h i e f e k t y w n i e j s z y c h systemów, a także o znalezienie skuteczniejszych od dotychczasowych s p o s o b ó w wykonywania prac projektowych.

Realizacja tych postulatów wytyczy n o w e k i e r u n k i w ASIZ, które są zdeterminowane przede wszystkim zmianami w technologii przetwarzania danych /transmisja danych, zarządzanie danymi, metody wejścia-wyjścia i in./, metodyce tworzenia systemów oraz metodach i technikach analizy.

3. Wybrane metody w analizie systemów informatycznych i sposoby ich wykorzystania

Zdaniem wielu autorów ASI jest rozszerzeniem dyscyplin tradycyjnego rozwiązywania problemów, które są stosowane w praktyce jednostek gospodarujących /obiektów/. Dyscypliny te najczęściej służą:

a/ określaniu systemów i ich otoczenia,

b/ przedstawianiu systemów,

c/ analizie i optymalizacji ich struktury,

d/ analizie i optymalizacji ich zachowania.

Stąd też w ASI znajdują zastosowanie metody i narzędzia wykorzystywane przez ekonomię, matematykę, organizację i zarządzanie, psychologię, socjologię i inne dyscypliny. Zaistniała więc potrzeba p o ł ą c z e n i a tych elementów i stworzenia o d r ę b n e g o podejścia, jak najlepiej przystosowanego do rozwiązywania złożonych problemów doskonalenia funkcjonowania jednostek gospodarujących. W ten sposób powstała metodologia badań SI, obejmująca m.in. metody:

1/ określania celu i funkcji systemu,

2/ realizacji systemu,

3/ określania zbiorów informacji, na których system będzie działał,

4/ określania sposobów dystrybucji informacji,

5/ ustalania struktury zbiorów danych,

6/ wyboru procesu przetwarzania danych i właściwego wyposażenia technicznego.

Zastosowanie tych metod w ASI pozwala opracować z a l e c e n i a praktyczne, które powinny stać się punktem wyjścia projektowania tych systemów.

Innym zagadnieniem jest o r g a n i z a c j a i ustalenie z a k r e s u prac analitycznych, które kończą się opracowaniem "planu działania". Chodzi tu przede wszystkim o: sposoby ustalania faktów /np. obserwacje, wywiad, ankieta/, środki służące do ich prezentacji /np. schematy blokowe, sieci przepływu danych, schematy czynności, tablice krzyżowe, tablice decyzyjne, modele sieciowe i in./, harmonogram prac analitycznych oraz metody ASI. Te ostatnie różnią się w zależności od celów stawianych ASI i możliwości przed ASI.

Przyjmując za kryterium mierzalność i niemierzalność materiału identyfikującego SI wyróżnia się i l o ś c i o w ą i j a k o ś c i o w ą ocenę poszczególnych elementów składowych SI. Podejście to umożliwia analizę k o m p l e k s o w ą, a więc ł ą c z n e poznanie czynników determinujących mechanizm funkcjonowania tych systemów, i dlatego znane jest jako "i n w e n t a r y z a c j a kompleksowa systemu informacyjnego", czyli IKSI /szerzej zob. [9]/.

W podejściu i l o ś c i o w y m stosowanym w ASI kwantyfikacja faktów zebranych w toku identyfikacji nie sprawia większych trudności. Inaczej jest w podejściu j a k o ś c i o w y m, które wymaga bardziej skomplikowanych i sformalizowanych reguł postępowania, np. opisu macierzowego. Opis macierzowy służy zarówno do analizy poszczególnych elementów SI w odniesieniu do obiektów /tablice typu I/, jak i do analizy tych elementów w obrębie SI.

W analizie przeprowadzonej według tej procedury wyróżnia się cztery fazy:

I - Analiza zbiorów informacji, w tym analiza:
  a/ funkcji, zbiorów informacji,
  b/ czynności przetwarzania zbiorów informacji,
  c/ transformacji zbiorów informacji,
  d/ powiązań nośników informacji.
II - Analiza kanałów, nadawców i odbiorców informacji,
III - Analiza środków technicznych przesyłania i przetwarzania informacji,
IV - Analiza jednostek funkcjonalnych.

Analiza jakościowa SI służy przede wszystkim ocenie jego użyteczności w procesie funkcjonowania. Osiągnięcie tego celu wymaga zastosowania dwojakiego typu tablic dwudzielnych.

Tablica typu I służy analizie poszczególnych elementów SI. Zawiera zatem w pionie: wszystkie pozycje analizowanego elementu SI występujące w danym obiekcie, a w poziomie: szczegółową charakterystykę tych pozycji. Tablica typu II dostarczająca natomiast materiału do analizy kompleksowej danego elementu, ale w odniesieniu do całego obiektu, zawiera w pionie: pozycje danego elementu, występujące w określonym obiekcie, a w poziomie: jednostki organizacyjne obiektu.

Szczegółowy opis podejścia jakościowego w ASI zawarty jest w [9]. Analityk posługujący się tą metodą uzyskuje informacje do podjęcia decyzji, których realizacja prowadzi do u s p r a w n i e n i a SI. Wynika stąd, iż analiza kompleksowa znajduje zastosowanie w badaniu funkcjonujących SI, nie zaś systemów będących w fazie projektowania bądź też systemów, które dopiero mają powstać. Konieczne jest zatem poznanie i upowszechnienie metody uniwersalnej, służącej wszystkim wymienionym celom. Metody takie są znane i z powodzeniem stosowane w krajach anglosaskich /por. m. in.[1],[2],[3],[4]/.

Jedną z takich metod opracowali J.E. Bingham i G.W.P. Davies /zob.[1]. Wyróżnili oni w niej sześć następujących faz:

I - wybór projektu systemu,
II - studium zastosowania systemu,
III - definiowanie systemu,
IV - projektowanie systemu,
V - wdrażanie systemu,
VI - ocena systemu.

Metoda ta jest zamkniętym cyklem badawczym "od projektu do projektu". Przydaje się zwłaszcza analitykowi nowicjuszowi. Punktem wyjściowym jest w y b ó r p r o j e k t u systemu, a ściślej identyfikacja zakresu badania, pomimo że faza ta jest zazwyczaj realizowana przez pion organizacji i zarządzania. Przed wybraniem projektu systemu należy określić, s p o ś r ó d jakich projektów nastąpi selekcja. Konieczne jest zatem uwzględnienie przez analityka najważniejszych źródeł p o m y s ł ó w i różnych rozwiązań problemu, przydatnych do rozważań nad projektem systemu.

Liczba rozwiązań i pomysłów zazwyczaj przerasta potrzeby i możliwości analityka, toteż można ją ograniczyć do tych, na które powinien on zwrócić baczniejszą uwagę. Chodzi tu o:

1/ zapotrzebowanie na informacje z innych działów,

2/ możliwości, jakie stwarza nowy sprzęt, oprogramowanie i techniki,

3/ pomysły wynikające z uprzednich badań nad systemem,

4/ możliwości powiązania lub integracji z innymi podsystemami,

5/ źródła zewnętrzne.

Sugestie i wymagania innych działów lub komórek stanowią istotne, często wykorzystywane źródło pomysłów dla projektowania SI. Sugestie te są tym bardziej cenne, że pochodzą od przyszłych użytkowników systemu, którym należy zapewnić odpowiedni serwis informacyjny. Są więc dobrą sposobnością do weryfikacji zawartości wejścia i wyjścia.

Postęp w produkcji sprzętu i oprogramowania na ogół stwarza nowe możliwości rozwiązania problemu, np. optyczne rozpoznanie pisma ręcznego lub maszynowego, potanienie transmisji danych. Należy jednak unikać przyjmowania z góry określonego rozwiązania zanim nastąpi sformułowanie problemu. Może to bo-

wiem doprowadzić do opracowania systemu, który bądź n i e z a s p o k o i potrzeb informacyjnych użytkowników, bądź też n i e r o z w i ą ż e ich problemów.

Sugestie i pomysły z poprzednich badań nad systemem są g ł ó w n y m źródłem pomysłów wykorzystywanych do opracowania nowego projektu systemu. Różne koncepcje i pomysły z poprzednich badań nad systemem zwykle pozostawia się i poszukuje lepszych rozwiązań. Stąd tak ważne jest utrwalenie wszystkich tych sugestii w odpowiednim czasie, by w momencie sprzyjającym /np. gdy użytkownik dysponuje już odpowiednimi funduszami lub gdy zostały spełnione inne warunki/ pogłębić badania nad systemem.

Podobnie jest z możliwościami połączenia lub integracji z innymi podsystemami. Podczas opracowywania projektu SI znacznie łatwiej jest zaobserwować istnienie powiązań /połączeń/ z innymi podsystemami. Czasami tych powiązań jest tak dużo i są tak złożone, że najlepiej byłoby rozpatrywać te podsystemy jako j e d e n zintegrowany system.

Wymiana poglądów, pomysłów i sugestii z innymi przedsiębiorstwami i organizacjami bezpośrednio lub poprzez publikacje, seminaria i organizacje profesjonalne jest opłacalne dla w s z y s t k i c h zainteresowanych i powinno stać się owocnym źródłem zewnętrznym pomysłów w opracowywaniu projektu SI.

Przedstawione w skrócie najważniejsze źródła idei i pomysłów dla wyboru projektu SI n i e s ą materiałem wystarczającym dla wykonania tego zadania. Uzupełnia się go zazwyczaj k r y t e r i a m i wyboru SI. Za najważniejsze z nich można uznać:

1/ możliwości zwrotu nakładów poniesionych na opracowanie SI /jest kryterium kluczowe/,

2/ potrzeby i wymagania różnych szczebli zarządzania,

3/ zastosowalność techniczna,

4/ zdolność zespołu lub komórki EPD do opracowania projektu SI,

5/ wymagania integracyjne z innymi systemami,

6/ potrzeby krytyczne przedsiębiorstwa.

Hierarchia ważności wymienionych kryteriów zależy od osądu doświadczonego zespołu lub komórki, która opracowuje systemy informatyczne zarządzania /SIZ/. Zasadność tego osądu jest we-

ryfikowana w toku późniejszego badania zastosowalności SI. Faza wyboru SI kończy się zdefiniowaniem c e l ó w projektów SI zakwalifikowanych do realizacji. Jest to istotne dla właściwego wykonania czynności w fazie następnej - studium zastosowalności SI.

S t u d i u m z a s t o s o w a l n o ś c i SI wskazuje dostępne, możliwe drogi realizacji celu projektu SI poprzez ustalenie nakładów i efektów każdego podejścia projektowego.

Studium zastosowalności systemu powinno przyczynić się do zaprojektowania jednego lub kilku rozwiązań szkicowych, dających odpowiedzi na następujące pytania:

1/ czy nowy system może zaspokoić cele ustalone przez kierownictwo obiektu?

2/ jakie korzyści i zalety będzie miał nowy SI w porównaniu ze starym?

3/ jakie będą koszty opracowania nowego SI?

4/ jakie będą koszty eksploatacji nowego SI?

5/ jakie będą koszty opracowania i eksploatacji nowego SI w świetle różnych celów funkcjonalnych?

6/ realizacja którego celu zapewni najważniejszy zysk?

7/ czy można sformułować program etapowego wdrażania SI, zapewniającego osiąganie najpilniejszych celów?

Uzyskanie odpowiedzi na powyższe pytania poprzedza wykonanie wielu złożonych czynności, do których należy zaliczyć:

1/ określenie charakterystycznych cech nowego SI,

2/ określenie wymagań na wyjściu łącznie z czasami odpowiedzi,

3/ zbadanie struktury organizacyjnej i przestrzennej obiektu,

4/ określenie struktury, zróżnicowania i przekrojów danych,

5/ rozważenie alternatywnych sposobów zaspokajania wymagań użytkownika,

6/ przeanalizowanie innych SI realizujących podobne wymagania,

7/ oszacowanie spodziewanych kosztów wdrożenia i eksploatacji SI dla każdej alternatywnej możliwości,

8/ oszacowanie efektów bezpośrednich i pośrednich dla każdej alternatywnej możliwości.

Najważniejszą częścią raportu studium zastosowalności SI jest analiza kosztów i efektów, składająca się z trzech części: koszty, efekty oraz ustalenie okresu zwrotu poniesionych nakładów inwestycyjnych. Szczegółowe rozważenie standardowej metody badania efektywności SI znajdują się w literaturze [8]. W wyniku tej analizy użytkownik dowiaduje się nie tylko o tym, czy wybrany projekt SI zaspokoi jego potrzeby informacyjne, ale także, czy jest to "przedsięwzięcie" zyskowne, stwarzające podstawy finansowe dalszego rozwoju zastosowań informatyki w przedsiębiorstwie, czy też odwrotnie - wdrożenie wybranego projektu SI przysporzy mu niespodziewanych strat. W przypadku drugiej ewentualności należałoby środki potrzebne na opracowanie, wdrożenie i eksploatację SI ulokować w innym przedsięwzięciu, dającym pozytywny wynik finansowy. Negatywny raport studium zastosowalności SI z a m y k a drogę do następnej fazy, czyli definiowania SI.

W odróżnieniu od wyboru projektu i studium zastosowalności SI f a z a d e f i n i o w a n i a jest badaniem bardziej szczegółowym, w wyniku którego ustala się, c o z a w i e r a SI. Kluczem do tej fazy jest ustalenie c e l ó w ogólnych i szczegółowych SI w razie zastosowania procedury krokowej /rozbicie celów ogólnych SI na cele szczegółowe podsystemów z zachowaniem ich wewnętrznej zgodności/. Z pozostałych przedsięwzięć umożliwiających zdefiniowanie zawartości SI należy wymienić:

1/ określenie danych elementarnych /najniższy poziom postaci logicznej jednostki informacji/,

2/ specyfikacja wejść i wyjść,

3/ gromadzenie danych i opracowanie zawartości zbiorów.

Efektem końcowym tych prac jest:

1/ zdefiniowanie celów systemu,

2/ uwzględnienie ograniczeń systemu /np. niektóre komórki organizacyjne przedsiębiorstwa mogą nie być objęte SI/,

3/ zdefiniowanie funkcji SI,

4/ opracowanie ogólnego schematu przepływu informacji,

5/ wyszczególnienie danych wejściowych, sposobów i środków ich pozyskiwania,

6/ wyszczególnienie danych wyjściowych i ich dystrybucja,

7/ określenie zawartości zbiorów i metod ich aktualizacji,

8/ podział obowiązków podczas wczytywania danych, ich aktualizacji, poprawy błędów itp.,

9/ ustalenie ograniczeń czasowych /ostateczny termin dostarczenia informacji użytkownikom systemu//.

Po zdefiniowaniu SI analityk może określić czas i środki niezbędne do zaprojektowania i wdrożenia systemu. Wynikają stąd naturalne związki między fazami: definiowania SI, projektowania i jego wdrażania. Powiązania te pozwalają na wprowadzenie ewentualnych poprawek po dokładnym zaopiniowaniu "marszruty" tych trzech faz.

W fazie projektowania analityk przechodzi od zawartości nowego SI /czyli od "co" do sposobów zaprojektowania i wdrożenia SI /czyli do "jak". Analityk podąża więc za uzgodnionymi c e l a m i i w y m a g a n i a m i do szczegółowej k o n c e p c j i SI, która ma być wdrożona. Ponadto w fazie tej kalkuluje się koszty eksploatacji SI i sporządza plan jego wdrożenia. Oznacza to, iż cały system musi być zdefiniowany w zakresie:

1/ przepływu informacji,

2/ zbiorów,

3/ woluminów,

4/ wyjścia,

5/ procedur,

6/ wzorów dokumentów wypełnianych ręcznie,

7/ specyfikacji programów itp.

Praca projektowa analityka SI nie jest łatwa. Powinien zatem on mieć nie tylko dużą wprawę i doświadczenie, lecz także znajomość i umiejętność stosowania nowoczesnych technik projektowania. Uwaga ta dotyczy głównie projektowania strukturalnego, umożliwiającego hierarchiczną dekompozycję problemu ogólnego, gdzie poziom niższy jest poziomem wyższym dla następnego, np. technika HIPO. Podejście "top-down" stwarza większe możliwości niż podejście modularne, które nie wskazuje tych samych, hierarchicznych związków między nimi. Do dyspozycji analityka systemu jest także podejście parametryczne, tzw. parametryzacja SI. Polega ono na tym, iż moduł o głównym przeznaczeniu jest tak zaprojektowany, iż może realizować różne cele w ramach istniejących rozwiązań /tailoring of the system/.

Konieczny jest wówczas odpowiedni zestaw danych w systemie zanim rozpocznie się dany przebieg /"ustawiony" na określone wykonanie poprzez pytanie/. Podejście parametryczne okazało się szczególnie przydatne w sytuacjach, kiedy trudno jest z góry określić szczegółowo potrzeby informacyjne użytkownika.

Spośród innych pomocniczych narzędzi projektowania dostępnych analitykowi, należy wymienić:

1/ sprawdzenie działania innych SI w podobnych sytuacjach,

2/ wykorzystanie dostępnych technik /np. schematy blokowe, metody statystyczne, symulacja/,

3/ zorganizowanie sesji "burzy mózgów", Sessions,

4/ wykorzystanie doświadczeń "Peer Group".

Zorganizowanie sesji burzy mózgów i wykorzystanie doświadczeń Peer Group zmusza analityka do jasnego i zrozumiałego przedstawienia jego podejścia w projektowaniu. Tym samym proces projektowania zostaje poddany sprawdzeniu przez osoby najlepiej zorientowane. Powoduje to nie tylko przepływ informacji o bieżącym zaawansowaniu prac projektowych w przekroju komórek i sekcji z tym związanych, lecz także przepływ wiedzy o technikach i narzędziach projektowania. Ma to szczególne znaczenie w odniesieniu do technicznych aspektów projektowania, których analityk nie jest w stanie rozwiązać.

Analityk przestaje uczestniczyć w fazie projektowania, gdy sporządzi specyfikację zawierającą:

1/ opis funkcji systemu,

2/ ogólny schemat blokowy ilustrujący główne czynności wykonywane w SI /manualnie i maszynowo/,

3/ identyfikację i powiązania wejścia, wyjścia i zbiorów,

4/ identyfikację i powiązania programów,

5/ schematy szczegółowe ilustrujące interakcje wejścia, wyjścia i zbiorów,

6/ specyfikację programów,

7/ specyfikację zbiorów.

Specyfikację tę uzupełnia się o szacunkowe koszty poniesione w fazie projektowania i wdrażania SI. Jeśli kierownictwo obiektu zaakceptuje projekt, wówczas nadaje się on do wdrożenia.

Przedmiotem f a z y w d r a ż a n i a jest doprowadzenie SI do stanu nadającego się do eksploatacji. Osiągnięcie te-

go stanu jest poprzedzone różnymi czynnościami. Warto zwrócić uwagę zwłaszcza na następujące:

1/ opracowanie i uruchomienie wszystkich programów,

2/ utworzenie kartoteki głównej,

3/ opracowanie dokumentacji dla komórki EPD /lub OO ETO/ i komórek użytkownika,

4/ zakup niezbędnego sprzętu i materiałów eksploatacyjnych,

5/ przeszkolenie pracowników komórki EPD /lub OO ETO/ i personelu użytkownika w zakresie eksploatacji SI,

6/ sprawdzenie modułów poszczególnych podsystemów,

7/ sprawdzenie i uruchomienie całego systemu.

Z zestawienia czynności wykonywanych pod nadzorem analityka wynika, iż trudno przecenić jego rolę w fazie wdrażania. Tezę tę potwierdzają liczne przykłady, z których wynika, że nawet dobrze zaprojektowane SI nie funkcjonują prawidłowo. Bywa nawet gorzej, niektóre z nich w ogóle nie są wdrażane bądź są wdrażane bez właściwego nadzoru. Problemy te występują nie tylko w naszym kraju, o czym świadczą badania przeprowadzone w Stanach Zjednoczonych, Anglii i we Francji. Autorzy wspomnianych badań stwierdzają zgodnie, iż do najczęstszych konsekwencji zaniedbań fazy wdrażania jest utrata zaufania użytkownika do SI. Dzieje się tak dlatego, iż kartoteki główne są niepewne, programy są uruchamiane bez uprzedniego usunięcia błędów, a użytkownicy nierzadko nie wiedzą, jak obsługiwać i eksploatować system. Wspólną przyczyną wszystkich tych zaniedbań jest zbyt krótki czas, jaki przeznacza się na testowanie SI. Nie trzeba nikogo przekonywać o tym, iż zaistnienie takich sytuacji stwarza poważne zagrożenie dla pomyślnego wdrożenia projektu SI, co gorsza, zagrożenie to przenosi się zwykle na projekty SI przewidzianych do wdrożenia w przyszłości.

Aby choć częściowo zapobiegać tym sytuacjom zaleca się analitykowi:

1/ stosowanie planowania formalnego i technik kontroli /np. wykresy Gantta, wykresy Gaspa, schematy czynnościowe, grafy/,

2/ opracowanie zestawu celów dla każdego członka zespołu projektowo-programistycznego, gdyż cele ogólne projektu nie są wystarczające. Istotne jest również wyznaczenie przez głównego projektanta realnych terminów zakończenia wyznaczonych zadań.

W przeciwnym razie czynności te mogą zbytnio rozciągnąć się w czasie i projekt nie zostanie ukończony,

3/ upewnienie się, czy wszystkie fazy projektowania zostały przedsięwzięte zgodnie z obowiązującymi standardami /analityk powinien zadecydować jakie standardy i gdzie należy zastosować/.

Spośród prac podejmowanych przez analityków w fazie wdrażania ważne jest przygotowanie użytkownika, a nierzadko również personelu fachowego EPD do pracy z nowym systemem. Dotyczy to przygotowania wejścia, poprawy błędów, obsługi komputera, rozdziału prac, iterpretacji wyjścia, wyboru materiałów eksploatacyjnych /w tym - materiałów piśmiennych i wzorów dokumentów/, wprowadzenia kontroli ogólnej, dystrybucji wydruków, wymaganych czasów itp.

Analitycy powinni dopilnować, by system wszedł do eksploatacji, gdy zostaną usunięte wszelkie usterki i błędy, a użytkownicy zapoznają się z jego obsługą. Wymaga to długiego testowania SI, moduł po module, a następnie testowanie całego SI z zastosowaniem danych próbnych i danych rzeczywistych.

Ponadto, analitycy powinni towarzyszyć użytkownikowi aż do czasu rozpoczęcia ciągłej eksploatacji SI. Tym sposobem użytkownik zyskuje wsparcie fachowe i psychologiczne. Ma to istotne znaczenie dla osób obsługujących system po raz pierwszy. Praca analityka związanego z danym projektem SI nie powinna ustawać w momencie rozpoczęcia eksploatacji i usunięcia błędów systemu. Niestety, zbyt wielu analityków uznaje swoją rolę za zakończoną na tym etapie i przechodzi do nowych projektów nie podejmując nigdy tak niezbędnej oceny projektów już wdrożonych.

F a z a o c e n y rozpoczyna się po kilku cyklach bezbłędnej pracy SI. Na przykład system o miesięcznym cyklu pracy należy oceniać po sześciu miesiącach jego funkcjonowania. Czas ten jest potrzebny do upewnienia się, czy użytkownik i personel EPD jest dostatecznie obeznany z systemem. System o tygodniowym cyklu obliczeń nie powinien być oceniany wcześniej niż po upływie dwu miesięcy eksploatacji, a system dzienny - po czterech tygodniach.

Istnieją co najmniej trzy powody, dla których dokonuje się oceny systemu. Po pierwsze, bada się jego efektywność, by u-

stalić, gdzie można ją zwiększyć. Po drugie, dokonuje się porównania osiągnięć SI z celami pierwotnie ustalonymi. Po trzecie, dzięki poznaniu zalet i wad SI analityk powiększa swoją wiedzę o przydatności systemu /swoiste sprzężenie zwrotne/.

Podsumowując post-ocenę analityk powinien zbadać funkcjonowanie SI w świetle następujących kryteriów:

1/ aktualne koszty,

2/ uzyskane efekty,

3/ tempo pracy systemu,

4/ stopień zaspokojenia potrzeb informacyjnych użytkownika,

5/ liczba błędów i stopień zawodności systemu,

6/ obszary problematyczne.

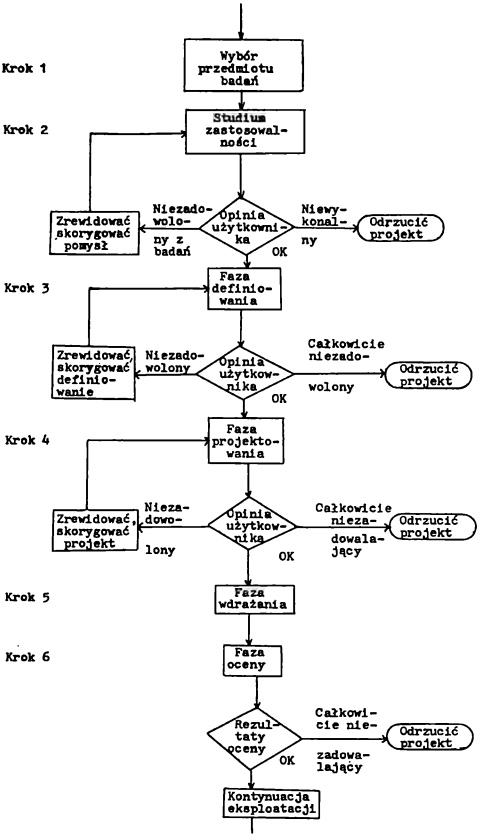
Wyniki tych badań powinny być porównywane z analogicznymi przewidywaniami poczynionymi w fazie zastosowalności i w fazie projektowania. Znaczące odchylenia należy poddać dalszym badaniom przyczynowo-skutkowym w celu ustalenia obszarów SI wymagających usprawnień, nierzadko dzięki drobnym zmianom.

O rezultatach oceny powinni być poinformowani zarówno użytkownik, jak i kierownictwo EPD /w specjalnym raporcie/. Czasem zdarza się, iż eksploatacja SI nie przysparza użytkownikowi zamierzonych efektów. Sytuacja taka może prowadzić do zawieszenia jego działania i zapobieżenia dalszym stratom.

Po wstępnej ocenie SI powinien być poddawany s y s t e m a t y c z n e j ocenie po rocznej eksploatacji. Trzeba jednak pamiętać, iż wiele systemów ulega stałym modyfikacjom wskutek zmian organizacyjnych, ekonomicznych i innych. Są to główne przyczyny utrudniające realizację pierwotnych celów SI. Okresowa ocena SI powinna dostarczać użytkownikowi informacji o tym, czy SI stale realizuje swoje cele, a także czy cele pierwotne są jeszcze aktualne.

4. Podsumowanie

Oddzielne rozpatrywanie każdej z sześciu faz analizy SI /według Binghama i Daviesa/ nie oznacza braku wzajemnego ich powiązania. W praktyce przejście do następnej fazy jest uzależnione od pomyślnego zakończenia poprzedniej. Wyodrębnienie poszczególnych faz daje analitykowi SI korzyści. Pozwala bowiem utworzyć węzły decyzyjne /zwane również węzłami kontrol-



Rys. 1. Powiązania między fazami analizy systemów informacyjnych i identyfikacją węzłów kontrolnych projektu /według [1, s. 39]/

nymi/, dostarczające informacji do podjęcia decyzji o kontynuacji bądź zaprzestaniu dalszego projektowania lub eksploatacji SI. Powiązanie i funkcjonowanie tych węzłów ilustruje rys. 1.

W przeglądzie wymienionych faz ASI warto też zwrócić uwagę na fakt, iż żadna z nich nie odnosi się do "analizy systemu i s t n i e j ą c e g o dotychczas". Ta ostatnia jest domeną analityków stosujących t r a d y c y j n e podejście z zakresu "organization and method". Nie oznacza to, iż poznanie dotychczasowego systemu jest zbędne. Należy jednak pamiętać, że projektowanie nowego SI nie może być nadmiernie uwarunkowane rozwiązaniami starego systemu. Analityk powinien wykazać duże zdecydowanie i wybrać najlepszą z dróg prowadzących do osiągnięcia celów systemu oraz uwzględnić metody i techniki aktualnie stosowane. Stanowisko to pozostaje w sprzeczności z tradycyjnym podejściem, streszczającym się w pytaniu: "jak można usprawnić istniejący system"?.

Ogólnie można stwierdzić, iż wiele SI powstaje w wyniku podejścia e w o l u c y j n e g o. Stąd też systemy te są w większym stopniu zdeterminowane przez wymagania przeszłości niż obecne. Eksploatuje się je i używa nowoczesne, szybkie i kosztowne systemy cyfrowe oparte jednak na przestarzałych procedurach. Procedury te początkowo stosowano w systemach cyfrowych bazujących na kartach dziurkowanych, następnie na taśmach magnetycznych i dyskach. Obecnie zaś - bez uprzednich, dogłębnych zmian - są eksploatowane w warunkach wieloprogramowości, transmisji danych i innych, nowoczesnych technologii przetwarzania danych.

Jest to klasyczny przykład podejścia tradycyjnego, w pełni zdeterminowanego systemem istniejącym. Zastosowanie tego podejścia zazwyczaj nie powoduje zmian organizacyjnych, kadrowych i w polityce przedsiębiorstwa.

Inaczej jest w podejściu, które scharakteryzowano w punkcie 3 niniejszego opracowania. Cechuje się ono całkowicie nowym spojrzeniem na rzeczywiste wymagania użytkownika. W wyniku tego nowy system jest zwykle inny od dotychczasowego i może powodować zmiany organizacyjne. Jest to zasadnicza przyczyna, z której powodu komórka EPD powinna przede wszystkim obsługiwać potrzeby decyzyjne a nie funkcjonalne w przedsiębiorstwie /np. obsługa rachunkowości/.

Trzeba jednak przyznać, iż dotychczasowy system jest p o d s t a w o w y m źródłem informacji o wymaganiach przedsiębiorstwa. Stąd pochodzą informacje i wskazówki dla każdej z sześciu faz analizy. Nie pomniejszając znaczenia tych prac nacisk powinien jednak być położony na definiowanie b i e ż ą c y c h wymagań.a nie poszukiwanie nowych sposobów zaspokojenia s t a r y c h potrzeb informacyjnych. W najbliższych latach powinien to być wiodący kierunek poszukiwań ASI.

[1] J.E. B i n g h a m, G.W.P D a v i e s, A Hand Book Systems Analysis, London 1978, The McMillan Press LTD.

[2] A. C h a n d o r, J. G r a h a m, R. W i l l i a m s o n, Practical Systems Analysis, London 1979, Rupert Hart Davies Educational Publications.

[3] H.D. C l i f t o n, Business Data Systems, London 1979, Prentice Hall International Inc.

[4] A. D a n i e l s, D. Y e a t e s, Basic Training in Systems Analysis, London 1979, National Computer Centre LTd.

[5] J. H a b r, J. V e p r e k, Systemowa analiza i synteza, Warszawa 1976, PWE.

[6] J. I l c z u k, M. J e r c z y ń s k a, Efektywność systemów informatycznych zarządzania, Warszawa 1979, PWE.

[7] J. K i s i e l n i c k i, Ekonomiczne problemy zautomatyzowanych systemów zarządzania, Warszawa 1981, PWE.

[8] B. K u b i a k, Metodologia oceny efektywności komputeryzacji, [w:] Informatyka w zarządzaniu przedsiębiorstwem, INFO-GRYF 83, Szczecin 1983.

[9] A. N o w i c k i, Analiza systemów informacyjnych, Akademia Ekonomiczna we Wrocławiu, Wrocław 1977, /skrypt/.

ON CERTAIN METHODS OF COMPUTER SYSTEMS ANALYSIS

Summary

The subject of the paper is a comparative analysis of Computer Systems Analysis, against the background of its possible uses in designing, exploiting and evaluating information and computer systems.

The author discusses the origin of Computer Systems Analysis and its development, particularly in the increased efficiency of management computer systems. He emphasizes the necessity for designing more efficient computer systems and defines the organization and range of analytical studies. Then he discusses the advantages of selected methods of Computer Systems Analysis.

The author proves the superiority of the so called "project to project" method, which is applied as early as that stage when the concept of the computer systems project is selected. The author pointes out the that greater stress should be placed on defining present information requirements instead of looking for new ways of satisfying former needs of computer systems users.

Sprzedaż skryptów i prac naukowych Akademii Ekonomicznej we Wrocławiu prowadzą księgarnie ORPAN: w Warszawie — Pałac Kultury i Nauki, 00-901 Warszawa, tel. 20-02-11, wewn. 21-05, we Wrocławiu — pl. Wolności 7, 50-071 Wrocław, tel. 340-16. Prace naukowe można zamawiać w kontynuacji, w miarę ukazywania się poszczególnych numerów, od najbliższego numeru aż do odwołania. Skrypty i prace naukowe można nabyć również w księgarni Domu Książki: 53-345 Wrocław, ul. Komandorska 118/120, tel. 68-11-55, wewn. 602